LORAL INSTRUMENTATION

FILE COPY
39224

SBA GPIB Application Note

SBA 100F

2-86 Release

8401 Aero Drive • San Diego, California 92123
TEL: (619) 560-5888

INTRODUCTION

This application note will overview the GPIB interface for the SBA product line, except the CIIL interface of the SBA-200A. All SBA's have GPIB interface capabilites, but the protocol varies from SBA-100/100A's and SBA-100F/200/200A's. This application note will overview the data transfers (records) for all SBA's, and the method of conducting GPIB data transfers for all SBA's. The organization of this note is as follows:

1. An overview of the GPIB standard as it applies to the SBA.
2. The Recommended GPIB Protocol Sequence.
3. SBA Record Formats.
4. Appendix A - ASCII Chart
5. Appendix B - Sample Program
6. Appendix C - Operation mode addresses.

The first section discusses the GPIB standard and the portions of the standard which the SBA supports, most of which is common to all GPIB devices. The second section overviews the recommended SBA-GPIB protocol. Section three presents the SBA records, and section four reviews the data structures for SBA-GPIB transmissions. The Appendixe are added to help speed along the writing of the host's driver to the SBA. The experienced GPIB programmer can probably skip the first section and just reference section 2 and 3. Sections and paragraphs will be numbered for easy reference.

Table of Contents

Section-Page	Topic
1-1	GPIB Overview
1.1-1	Background
1.2-3	Handshaking
1.3-3	Addressing and other GPIB Commands
1.3.1-4	GPIB Commands
1.3.1.1-6	GPIB Control or Management Lines
1.3.1.2-7	Serial and Parallel Polling
2-1	Recommended SBA-GPIB Sequence
3-1	Record Formatting
3.1-1	Records to SBA
3.1.1-1	Record Formats - Overview
3.1.2-2	Communication Encoding - SBA-100/100A Only
3.2-3	Command Records
3.3-4	Page Update Records
3.4-5	Memory Records
3.5-6	Header Records
3.6-7	Line Printer Records
3.7-8	Records from the SBA
3.7.1-8	Memory Records
3.7.2-8	Header Records
3.7.3-9	Stack Records
3.8-11	GPIB Record Review
3.8.1-12	Encoding for SBA-100/100A
3.8.2-14	Encoding for SBA-100F/200 and 200A
3.8.3-15	GPIB Comparision between SBA-100/100A and SBA-100F/200/200A
Appendix A	ASCII Chart
Appendix B	Sample Program
Appendix C	Operation Mode Addresses

Digitized by the Internet Archive
in 2024

https://archive.org/details/manualsplus_01626

1.0 THE GPIB STANDARD - AN OVERVIEW

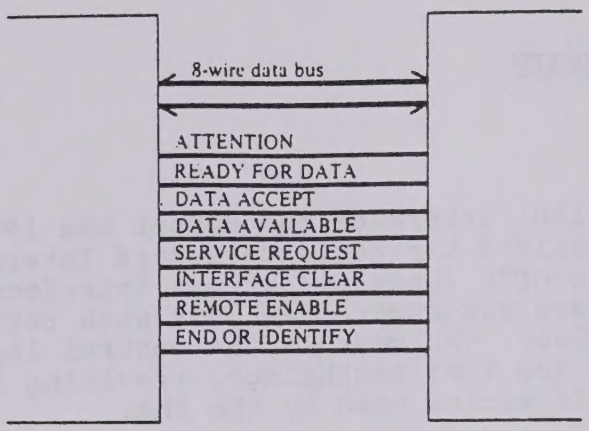
1.1 Background

With the need of a standard parallel interface throughout the 1960's and early 1970's, the IEEE standarized the Hewlett-Packard Interface Bus (HPIB) naming the IEEE-488 or GPIB (General Purpose Interface Bus). The original IEEE and ANSI standard was enacted in 1975 with revisions in 1978. With the standard in place, the cost of the control logic decreased rapidly because of LSI and VLSI technology, resulting in compact chip sets like the TI 9914 series used by the SBA.

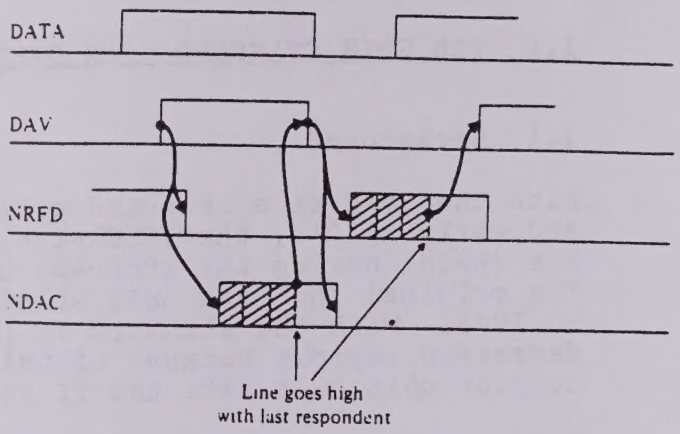
Today, most computer manufacturers implement or offer the IEEE-488 interface bus with their equipment, as does the SBA product line. The obvious advantage of the GPIB over the conventional RS-232 port is speed. Other advantages include control of many instruments or computers over a single bus, and pre-defined bus commands for common communications and a means of obtaining peripheral status. The GPIB bus has one Bus Controller (master) that initiates all bus commands and data transfers. The other instruments (slaves) on the bus may have capabilities of being the controller, but a slave may not act as the controller until the task is assigned to it (much like the MIL-STD-1553 bus). The SBA can NEVER be the master on the GPIB bus, only a slave. The GPIB bus uses negative logic where true is zero and false is one.

The GPIB bus is also known as the three wire bus, which in a handshake sense is true, but there are also five management/control lines and eight data lines (Data In Out - DIO). The standard also provides eight additional lines for shielding and ground. The handshake lines of the GPIB bus interface are: the Not Data Accept (NDAC) line; the Not Ready For Data (NRFD) line; and the Data Available (DAV) line. Figure 1-1 (next page) shows a timing diagram and Figure 1-2 shows a flow chart for a bus data handshake. Below is a figure of the pin-outs for the SBA-GPIB port.

SIGNAL	DB25S PINS
D101	1
D102	2
D103	3
D104	4
D105	13
D106	14
D107	15
D108	16
REN	17
IFC	9
NDAC	8
NRFD	7
DAV	6
EOI	5
ATN	11
SRQ	10
GND*	18-24
SHIELD*	12

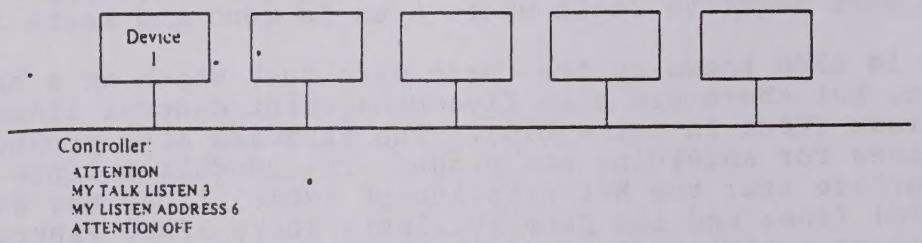


IEEE-488 bus configuration.

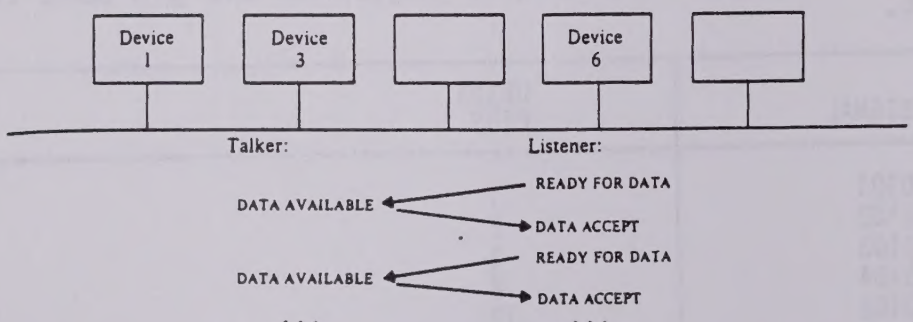


IEEE-488 3-wire data transfer handshake.

Figure 1-1



(a) First step in IEEE-488 I/O transfer.



(b) Data transfer on an IEEE-488 bus.

Figure 1-2

1.2 HANDSHAKE CONTROL

As seen in Figure 1-1 and 1-2, the data handshake of the GPIB is semi-complicated. Two common facts about transfers are: the NRFD and NDAC lines are never to be high (false) at the same time; and data will never be placed on the data lines (DIO) if NRFD and NDAC are high at the same time. The following steps overview a GPIB transfer as seen in Figure 1-2.

1. The master or source device senses for NRFD and NDAC high simulataneously. If the two lines are not high, data will be put onto the data lines.

2. The slave or listener should have NRFD high and NDAC low in a ready state. If this condition exists, the master will pull DAV low. This can be called the beginning of the transfer.

3. When DAV goes low the slave should accept the data and pull NRFD low.

4. The slave should release NDAC high when the data has been accepted forcing the master to also release the DAV line. The slave monitors the DAV line and pulls the NDAC line low when the DAV line is sensed high.

5. The slave keeps the NRFD line held low until it is able to assume a ready state. In the mean time, if more data is to be transfered, new data is presented on the DIO lines and the process starts again.

1.3 ADDRESSING and other BUS COMMANDS

As mentioned before, the GPIB bus has set bus commands (including addresses) to configure and monitor bus status. These commands are sent to slaves from only the master in the manner described in section 1.2. However, an additional line is included in the handshake for GPIB bus commands: the Attention line (ATTN). The master asserts (low) the ATTN line with the DAV line when a GPIB bus command is going out to a device(s). This way, the slaves on the bus can distinguish between a regular data transfer and a bus command. GPIB bus commands are reviewd in section 1.3.1.

The SBA has a set GPIB address which is changable on the SBA's I/O page (100 models) or from the front panel of 200 models (see the User's Manual for your respective machine for settings). The SBA will check each address presented on the bus to see whether or not it will participate in a transaction.

When the master of the GPIB (probably your host computer) is to initiate a data transfer, a master talk and SBA listen address must first be sent over the bus (probably want to preceed this with a Untalk/Unlisten). When the SBA has sensed its matching address, a data transfer from master to SBA may executed. Of course, if the SBA is to send data back to the master, the SBA would be addressed as the talker and the master would be the listener. Most computers have built-in GPI commands in their high-level software, and these commands automaticall send the proper GPIB commands and control the handshake lines (NDAV, NRFD, DAV, ATTN).

1.3.1 GPIB COMMANDS

The remainder of this section will review each of the 15 GPIB commands applicable to the SBA product line. The first group of commands are implemented by sending over a predefined data byte as explained in section 1.3. The second group (four of eight control lines) are initiated by asserting a separate line on the bus.

\+ DCL (DEVICE CLEAR)

This command causes the SBA to "RESET", putting all SBA I/O ports in an idle state. No SBA page variables are affected.

\+ * GET (GROUP EXECUTE TRIGGER)

This command causes Keyrecord 2 of the SBA to be executed.

\+ * GTL (GO TO LOCAL)

This command UNLOCKS the SBA keyboard and keypad.

\+ * LLO (LOCAL LOCKOUT)

This command LOCKS the SBA keyboard and keypad until an GPIB DCL (DEVICE CLEAR), or GTL (GO TO LOCAL) is sent to the SBA.

LISTEN

This command addresses the SBA or master as the LISTENER.

TALK

This command addresses the SBA or master as the TALKER.

\+ PPC (PARALLEL POLL CONFIGURE)

This command sets-up the SBA for a following parallel poll configuration command. A PPE (PARALLEL POLL ENABLE) or PPD (PARALLEL POLL DISABLE) must be sent immedietely after this command (PPC). See Section 1.3.1.2 for details.

\+ * PPE (PARALLEL POLL ENABLE)

This command defines the data line assigned to the SBA when a parallel poll is initiated on the GPIB bus. The SBA will assert the assigned line during a parallel poll if the SBA is in a "ready" state. See Section 1.3.1.2 for details.

\+ * PPD (PARALLEL POLL DISABLE)

This command clears the SBA assigned line for a Parallel Poll. See Section 1.3.1.2 for details.

\+ PPU (PARALLEL POLL UNCONFIGURE)

This command clears the SBA's parallel poll configuration, including any previous PPE command. See Section 1.3.1.2 for details.

\+ * SDC (SELECTIVE DEVICE CLEAR)

This command reinitiates the SBA to a power-up state, resetting all SBA variables to default values.

SPD (SERIAL POLL DISABLE)

This command clears a serial poll configuration and the data associated with it. See Section 1.3.1.2 for details.

SPE (SERIAL POLL ENABLE)

This command initiates a serial poll configuration of the SBA setting a byte in an internal register to be read by the master for status. See Section 1.3.1.2 for details.

UNL (UNLISTEN)

This command clears all slaves (including the SBA) to a

non-listen state so they will not accept data from the bus, enabling a new listen(ers) to be established.

UNT (UNTALK)

This command clears all slaves (including the SBA) to a non-talk state, enabling a new talker to be established.

KEY:

+ The SBA-100 and 100A models do not support these functions.

* These commands require that the SBA FIRST be addressed as the listener. All other commands are targeted for all devic

1.3.1.1 GPIB CONTROL or MANAGEMENT LINES

The previous sections have covered four of the eight GPIB control li (NDAC, NRFD, DAV, ATTN). This section will cover the remaining four

+ IFC (INTERFACE CLEAR)

  This line, when asserted by the master, clears or resets the SBA's GPIB controller chip. The interface is left in a ready state.

+ REN (REMOTE ENABLE)

  This line, when asserted by the master, forces the SBA into a GPIB ready state and locks the SBA's keyboard and keypad. This action requires the SBA first be addressed as listener.

EOI (END OR IDENTIFY)

This line is driven low with the last byte of the last recc transmitted by either the master or SBA. This standard GPI convention defines the end of a data transfer on the bus. Tl line is also used in conjuction with the Attention Line (Section 1.3) to initiate a parallel poll.

SRQ (SERVICE REQUEST)

This line is driven low by the SBA after executing a commanc record sent by the master, or when the SBA detects an inter failure upon power-up. The master would usually conduct a serial poll upon detecting this line asserted by the SBA, a

once the serial poll has been completed (SDP command) , this line returns to a normal state. See section 1.3.1.2.

+ SRQ - Output Status

The SBA-100F/200/200A models have the capability to force th SRQ line true, enabling a user defined status byte to be rea when a sreial poll is performed. The user may define a memor word of the SBA (4105H for SBA-100F/200A - 481DH for SBA-200 to any value and then execute an SBA OUTPUT STATUS command to assert the SRQ line true. A serial poll may then be performed to obtain the word defined. This feature could be very handy when using keyrecord programs and the host comput is to be informed of a given condition. See your User's Manual for more details.

KEY:

+ SBA-100 and 100A models do not support these functions.

1.3.1.2 Serial and Parallel Poll Overview

The serial and parallel poll commands of the GPIB allow the master to inquire the status of devices on the bus. However, each technique of inquiring has its advantages and disadvantages. The serial poll technique provides the means of aquiring the status condition of a single device. The parallel poll technique allows the master to inquire about the "ready" condition of several devices at the same time. The preferred technique is the serial poll because more information is conveyed.

Serial Polling the SBA

A serial poll is conducted on the SBA to obtain a status byte from th SBA. This byte is loaded into a register on the SBA GPIB controlling chip by the SBA operating system. The SBA operating system monitors the status of the GPIB interface and updates the serial poll register when a GPIB SPE command transmitted.

The following sequence is a typical GPIB transaction of a serial poll of a slave.

1. The master sends an UNLISTEN command.
2. The master sends a SERIAL POLL ENABLE command.
3. The master sends the SBA TALK address.
4. The SBA sends the STATUS BYTE to the master.
5. The master sends the SERIAL POLL DISABLE command.

A serial poll should be conducted whenever the SBA asserts the SRQ line. The SBA will assert the SRQ line on two separate occasions.

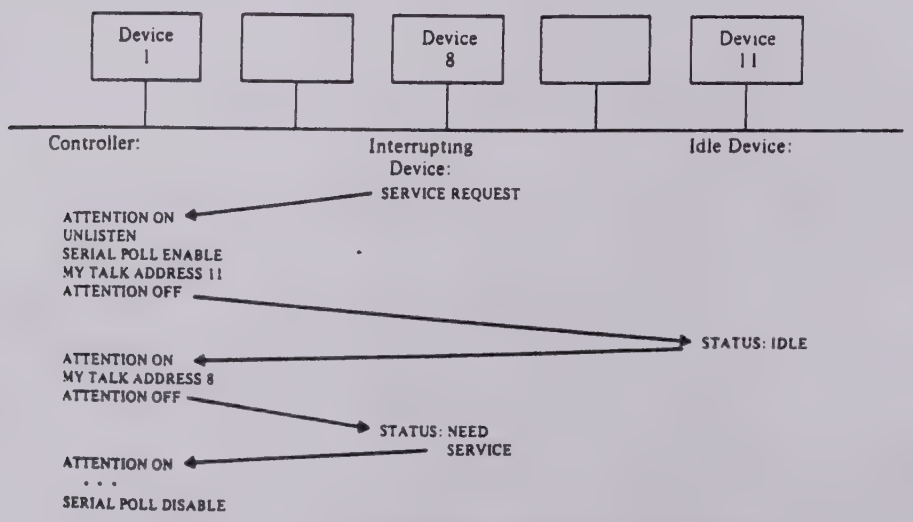
1. On the event of a board failure during a power-on syst test of the SBA.

2. After successfully completing execution of a command record sent to the SBA over the GPIB port.

In the event of a power-on failure, the status byte can be read to determine what type of failure occured. Asserting the SRQ line afte the successful execution of a command record over the GPIB port all the host to determine when the SBA is ready to accept a new record. The following is a list of the possible status bytes transmitted by SBA.

01000000	Command string completion-no error
11000001	Invalid LRC detected
11000011	CRT failure in test
11000100	BOP failure in test
11000101	BAP failure in test
11000110	BIP failure in test
11000111	DOP failure in test
11001000	Transmit test failure

The detection of a bad LRC brings different reactions by the differ models of the SBA. Upon detection of a bad LRC, An SBA-100/100A an 200 will not execute the record sent to the SBA. However, an SBA-1 or 200A will try to execute the record the best it can. In either c the status byte will contain the value for detection of a bad LRC (shown above).



(c) A typical IEEE-488 bus transaction for a service request (serial-poll mode).

Parallel Polling the SBA

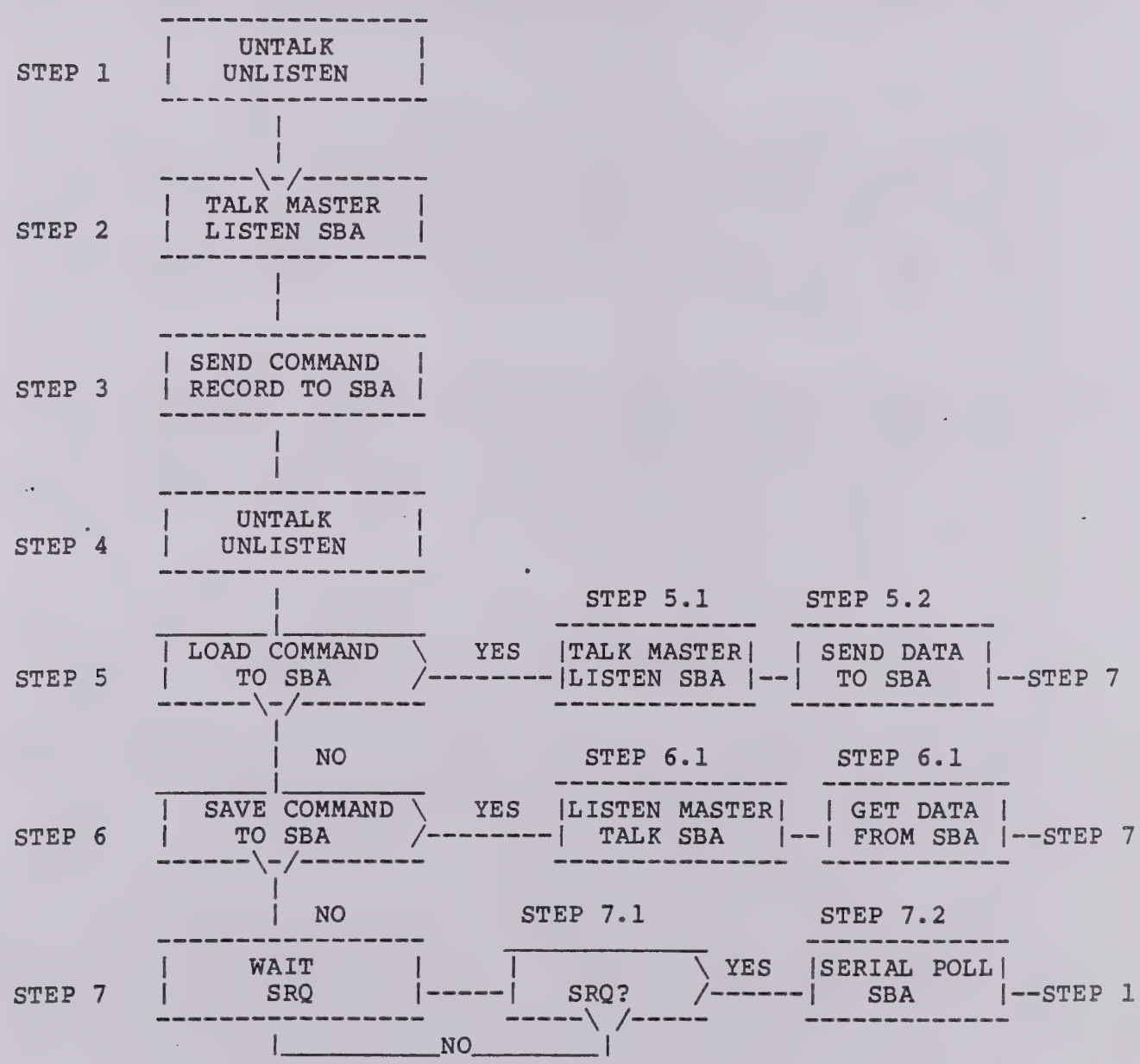
A parallel poll is conducted on the GPIB bus to obtain the ready condition of several devices. A maximum of eight devices can be polle at one time. Each device is assigned a data line (one of eight data lines of the bus) on which to respond by a PPE command. The master of the bus sends a PPC command to all devices, and then sends a talk address and a PPE command to the devices that will participate when a parallel poll is initiated.

A parallel poll is initiated when the master asserts the Attention and EOI lines (see Section 1.3.1.3) simultaneously. The participating devices will pull their assigned data line to a true condition (one) i the device is in a ready state. The device will let the line float false (zero) if the device is not ready. In this manner, the master can determine the non-ready devices on the bus, and take appropriate action (usually a serial poll). Assigned data lines can be cleared by the PPD (for a particular device), or a PPU command (clears the bus's parallel poll configuration).

It should be clear that performing a parallel poll on a small system would take about as much time as a serial poll on each device, and the amount of information conveyed makes parallel polling inefficient.

2.0 THE RECOMMENDED SBA-GPIB TRANSMISSION SEQUENCE

This section will overview the recommended transmission protocol for all SBA units. Throughout this section, references of SBA records are given, so you may want to familiarize yourself with the different record types as discussed in section 3. A flow chart of the interface is given below, and a discussion of the sequence starts on the next page.



STEP 1.

This is the initial step in setting up a GPIB transfer. The GPIB controller should send the universal commands UNTALK and UNLISTEN to clear bus activity. This action insures that the bus is free and communications with the SBA will not be interrupted.

NOTE: If you are using an SBA-100 or 100A, an ASSIGN CONTROL TO GPIB command must be executed before ANY SBA-GPIB communications can take place.

STEP 2.

This step puts the SBA into a listen state. Send the MLA to the SBA and the MTA to the master. The SBA will remain in the listen state until an unlisten command is received. This action will cause the SBA keyboard to lock-out until GPIB communications are ended. Most high-level software "OUTPUT" commands automatically address the GPIB bus.

STEP 3.

The master must send a command record to the SBA. Command records are always the first record transmitted to the SBA. If the master is only transmitting command records, then the logic pathway may skip to step 7. If data records (page-update, memory, header, or line printer) are to be sent or received from the SBA, then a LOAD or SAVE command record must have been sent to the SBA.

STEP 4.

This step (untalk, unlisten) once again clears the the GPIB for further SBA record transfers, and in some cases is optional. If the master is through sending command records to the SBA, then this step can be ignored and the driver program can skip to Step 7. Other GPIB transfers probably should first include this step to insure clean communications over the GPIB. See step 1 for more details.

STEP 5.

This step is only a logical pathway. If the Command Record sent to the SBA in step 3 was a LOAD FROM GPIB command, then the logic pathway of the interface proceeds to step 5.1. If the interface has reached this point, then a data record (as given in step 3) is to be sent to the SBA.

STEP 5.1

This step addresses the SBA and master to listen and talk, respectively. This step is optional depending on the action taken in step 4. If the bus was not cleared (untalk, unlisten), then the SBA would still be in a listen mode and could continue to accept data from the host. If the bus was cleared then the MLA and MTA commands would need to be sent before sending a data record to the SBA. See step 2 for more details.

Step 5.2

In this step, the master is actually sending a data record(s) to the SBA. Upon receiving the record(s), the SBA will display on the CRT the memory or function which it has loaded. The record's data should be handshaked as descibed in section 1.2 (normal GPIB). The host must assert the EOI line with the last byte of the last record sent to the SBA. Goto to step 7 when all data has been sent to the SBA.

STEP 6.

This step is only a logical pathway. If the Command record sent to the SBA in step 3 was a SAVE SBA function TO GPIB command, then the logic pathway of the interface proceeds to step 6.1. Step 4 must have been performed to clear the SBA out of the listen state.

STEP 6.1

This step addesses the SBA to TALK and the master to LISTEN. This step initializes an SBA transfer to the master with the requested SBA function or data that was in the SBA-SAVE GPIB command.

Step 6.2

Upon being addessed as the talker in step 6.1, the SBA will transmit the desired data (in records) as specified in the SAVE command record in step 3. The SBA will either send requested header, stack or memory records to the host. The SBA sends all data in memory records except header or stack information. Handshaking is performed as descibed in section 1.2 (normal GPIB). The SBA will assert the EOI line with the last byte of the last record sent to the host.

STEP 7

This step is an interface wait state. Before a serial poll on the SBA can be performed, the master should wait for the SRQ line to be asserted. The SRQ line will be asserted by the SBA according to conditions descibed in section 1.3.1.3 (upon power-up board failure or after the completion of Command Record execution.)

STEP 7.1

This step is only a logic pathway. If the SRQ line assurtion is detected by the master then proceed to step 7.2 and perform a serial poll on the SBA. If the SRQ is not detected, then the driver program should wait until the condition is true before continuing.

STEP 7.2

This step has the master performing a serial poll on the SBA. The recommended protocol and SBA reactions are descibed in section 1.3.1.3. Once the serial poll has been performed, and further communications are still desired, return to step 1. If communications with the SBA has ended, the SBA should be cleared of all GPIB status (maybe assert IFC - section 1.3.1.2).

Review

The transmission sequence descibed in the preceeding sections outlines almost any interface with the SBA over the GPIB bus. The protocol to the SBA is not fixed, and other usual standard GPIB command sequences can be used.

Many computers today have built in GPIB functions where handshaking and addressing are performed automatically. These features are usually not a problem if the speed of the GPIB commands are not contiguous (or close). Therefore it is sometimes preferable to single step the GPIB commands as descibed in the flow chart of this section. The serial polling of the SBA is usually where the automatic GPIB controlling computers have some problems. If these automatic functions are performed to quickly, then the SBA will not be able to read GPIB commands on the bus. The SBA-100/100A's and 200's are slower than the 100F's and 200A's.

3 Record Formatting

Records are formatted data strings which are transfered on the GPIB bus to and from the SBA. Each record has a minimal amount of data identifier and communication bytes to ensure transmission intrigrity. This section will overview the SBA record formats and pertaining identifier bytes for each record. Section 4 reviews communication formats and gives a comparison of the GPIB protocol of SBA the different SBA models.

3.1 Records to the SBA

Most SBA data records require four identifier bytes at the beginning of a record format and one record requires a termination byte (single) at the end of the data string. SBA-100's and 100A's also include a COMMUNICATION ENCODING of the records which is often used with modem or other secure data transmissions: an DLE-STX-ETX protocol. These added bytes will be referred to as COMMUNICATION ENCODING and will be described further in section 3.1.2 and section 3.8. SBA-100F/200 and 200A's DO NOT have this encoding; they have the record identifier and data encoding techniques which is common for all SBA's. Again, records for all SBA's have the same record encoding formats, but the SBA-100 and 100A's have ADDITIONAL COMMUNICATION ENCODING bytes.

3.1.1 Record Formats

As stated before, most records have four identifier bytes. The first byte (left-most) is an ID (identification) byte which distinquishes the type of record. The second and third byte are usually reserved for SBA internal address locations. And the fourth byte is a byte count variable for the record transfer. ALL SBA records must have an LRC (Longitudinal Redundancy Check) character which is an exclusive-ORing of ALL the bytes in a standard SBA record. An example for calculating the LRC character is given in the sample program.

The following sections will present record formats.

Command Records.....Section 3.2
Page Update Records.....Section 3.3
Memory Records.....Section 3.4
Header Records.....Section 3.5
Line Printer Records....Section 3.6
Stack Records.....Section 3.7.3

Section 3.7 will overview data records sent to the host from the SBA. Section 4 provides a review of record encoding.

3.1.2 Communication Encoding

If you have an SBA-100 or 100A, then there is an encoding of records which must be performed (if you do not have an SBA-100 or 100A, then this section does not apply). Besides the identifiers, termination and LRC characters, an DLE-STX-ETX sequence must be added. This sequence does not alter the format of identifiers or termination bytes, and the LRC character is still the very last byte of a record. The communication encoding is as follows, and the values for the characters are in HEX underneath:

DLE	STX	[Standard Record]	DLE	ETX	LRC
10	02		10	03	

All record transmissions to or from an SBA-100 or 100A must have this encoding scheme. None of the characters EXCEPT the ETX characters and those in the standard record will be included in the LRC calculation. The ETX character must be included in the LRC calculation. Remeber to include this encoding when reading the sections of record explanations (3.2-3.7).
This encoding scheme may alter the size of the records depending on the byte values in a standard record.

Further explanation is given in section 3.8.

3.2 Command Records

Command Records are actually SBA ASCII characters that directly correspond to normal keystrokes entered from the SBA keyboard. This allows the user to setup the SBA from a host computer exactly as if setting-up the SBA from the keyboard. Figure 3.1-1 shows the format for Command Records.

```
                ---------------------------------------
                |          ASCII CHARACTERS           | LRC
                ---------------------------------------

                ---------------------------------------
        i.e.    |    DB01<cr><lf>C01R01010001MPY<cr>  | LRC
                ---------------------------------------
                        <cr> = carriage return
                        <lf> = line feed
```

The example would setup BC page 01 and includes the normal carriage returns and line feeds (down arrow). Notice that only the significant characters need to be sent, just like keyboard action. Use an ASCII space character for right arrow. See Appendix A for ASCII characters. Use ONLY upper case ASCII characters. Maximum ASCII string length is 256 characters (not including the LRC byte or any communication encoding).

SRQ

Command Records are the only record, which upon SBA execution, cause the SRQ line to be asserted.

3.3 Page Update Records

This record allows the user to automatically update portions of the operation system of the SBA by directly transfering binary data. The pages that can be updated by this record transfer are: BC pages, RT pages, Frame Format page, Operation Mode 1 and 2 page, Capture page, Priority page, and I/O page. The memory locations and memory setup values are given in Appendix C. This record type allows a much faster way of setting up the SBA compared to a command string transfer.

When this record type is transmitted, the SBA stops transmitting any active BC's and updates the hardware with the new setup. The SBA will restart BC transmission upon the completion of the master's transfer. The following describes the record format, including the four bytes of identifiers.

ID	LSB	MSB	BC	BINARY DATA	LRC

ID = 01 - HEX
LSB = Least Significant Address Byte - HEX
MSB = Most Significant Address Byte - HEX
BC = Byte Count of BINARY DATA - HEX
BINARY DATA = Data Values Correspoding to the Address Location (LSB,MSB).

i.e.

01	00	50	04	4 binary bytes	LRC

This example would transfer the 4 binary bytes to address 5000-5003 and would automatically update the respective hardware.

The LSB and MSB location is the starting address for a memory load and the following data bytes will be loaded sequentially at that starting address. If a new address location is desired, a new record must be sent to the SBA. The maximum byte count per record is 256, including the identifier bytes, but not including the LRC or any communication encoding (see section 3.8).

3.4 Memory Records

Memory Records are used to transfer 8-bit values to SBA memory. These values could be used for page-updating (similiar to section 3.3), down-loading of 8080/85 machine code programs, or for moving CRT displays, etc. The format for memory records is shown below.

ID	LSB	MSB	BC	BINARY DATA	LRC

ID = 04 - HEX
LSB = Least Significant Address Byte - HEX
MSB = Most Significant Address Byte - HEX
BC = Byte Count of BINARY DATA - HEX
BINARY DATA = Data Values Correspoding to the Address Location (LSB,MSB).

i.e.

04	00	60	06	6 BINARY BYTES	LRC

This example would transfer 6 binary bytes to addresses 6000-6005.

The LSB and MSB location is the starting address for a memory load and the following data bytes will be loaded sequentially at that starting address. If a new address location is desired (other than sequential), a new record must be sent to the SBA. The maximum bytes per record transfer is 256, including the identifier bytes, but not including the LRC or communication encoding (see section 3.8).

Using this record for page-updating can be done, but extra steps must be made to force the SBA to update the system; this application for memory records is not recommended.

Most of the time, this record is used to down-load machine code programs or other 8-bit values to Keyrecord memory (514E-7000). See your User Manual for more details.

3.5 Header Records

This record allows a "Header" or ASCII string to be transferred to the top of the SBA's CRT for user defined messages. The message is the ASCII string and is formatted as follows:

ID	ASCII STRING	ID	LRC

ID = 03 - HEX

i.e.

03	SBA SETUP COMPLETE	03	LRC

This example would display SBA SETUP COMPLETE on the top of the CRT. The ASCII string can be no longer 37 characters. The entire record length can be no longer than 39 bytes including the ID, but not the LRC or communication encoding (see section 3.8). Use only upper case ASCII characters.

Zero Block Records

Zero Blocks are records which signify the end of a page-update, memory, header, and lineprinter records. The block consists of the respective identifier, address, byte count, and termination bytes for each record, but NO data (binary or ASCII) in the record. When the host (or SBA) is finished transmitting one of the records above, then a Zero Block record must be transmitted to conclude the transmission. The Zero Blocks for the each record are given below.

ID	LSB	MSB	BC	LRC
			00	

Page-Update and Memory Records

ID	ID

Header Records

ID	XX	XX	ASCII STRING	TM

Line Printer Records

See the respective section for byte values. Remember, the respective Zero Block record must sent after the transmission of the respective record(s). The Zero Block is always the last record. Command Records DO NOT require a Zero Block transmission.

NOTE: This page is an insert for 2-86 release Application Note. Incorporations to each record section will be made in a later release. Please call an Application Engineer is you have any questions.

3.6 Line Printer Records

A Line Printer Record is read by the SBA and the SBA directs an ASCII string from the GPIB to the SBA Line Printer port. The SBA does not change any characters of the ASCII string. The structure is as follows:

ID	XX	XX	XX	ASCII STRING	TM	LRC

ID = 02 - HEX
XX = Don't care - can not be ZERO
TM = 06 - HEX (Terminator)

i.e.

02	11	11	11	DATA PAGE ONE - 10/29/85	06	LRC

Be sure include all line feed and carriage return characters or any other control characters. The maximum ASCII string length per record is 256, including the identifier bytes, but not the LRC or communication encoding (see section 3.8). See your Users Manual for line printer information and Appendix A for a ASCII chart.

3.7 Records from the SBA

Records are sent to the SBA upon a SAVE function TO (>) GPIB command execution by the SBA. Once the SBA has been addressed to talk, it will transmit the requested data (function) in continuous records. The EOI line will be asserted with the LRC character of the last record. There are only three types of records that can be sent from the SBA: memory, stack, and header. These records will now be discussed.

3.7.1 Memory Records from the SBA

Memory records sent from the SBA are exactly as described in Section 3.4 where the LSB and MSB bytes represent the address location from which the SBA sent the information. The SBA sends all modes or functions, except stack and header strings, in memory records (see Appendix C for operation mode addresses). There are many uses for memory records, such as, reading operation mode addresses, CRT displays, etc. And by reading the addresses from the records, the data could be changed and sent back to the SBA for modifiying the SBA. Of course, if the operation mode is to changed, then a page-update record would be used (just change the ID byte to a page-update record and retransmit the record after modification).

3.7.2 Header Records from the SBA

Header records sent by the SBA are identical to Header Records sent to the SBA (see section 3.5). These records could be used for flagging the host computer about a certain condition in a Keyrecord program.

3.7.3 Stack Records from the SBA

The Stack Record is the only record that is unique to SBA transmissions. Upon the execution of the SBA command SAVE STACK XXXX,XXXX TO (>) GPIB, where XXXX,XXXX represents the starting and ending address locations, the SBA will transmit the requested stack data in a unique record(s). The SBA can only transmit stack data and can not load stack data (accept from host).

The SBA transmits a single record for each line of the stack requested in the SAVE STACK command. A Stack Record has six bytes of stack data for each line and four bytes of identifiers which are broken down as follows:

ID	LSB	MSB	BC	6 BINARY BYTES OF DATA	LRC

ID = CC - HEX
LSB = Least Significant Byte of Stack Memory - HEX
MSB = Most Significant Byte of Stack Memory - HEX
BC = 06H - Byte Count of the BINARY BYTES - HEX

The six binary bytes are encoded as follows.

STACK ADDRS LOW	STACK ADDRS HIGH	RECORD TYPE -	WORD DESCRIPTION	1553 STACK DATA BYTE	1553 STACK DATA BYTE

Stack Address (2 bytes) - LOW BYTE HIGH BYTE - HEX

Record Type (1 byte) - XXXXGXTC - HEX
XXXXXX = Don't Cares
If C = 1 This Stack Line is a New Command.
If T = 1 This Stack Line is a Time Tag.
If G = 1 The Time Tag is greater than 6.7 E 7.

Word Description - B EEEEE WW - HEX

B = 1 Stack Word on Primary Bus.
B = 0 Stack Word on Secondary Bus.

EEEEE = Error Type

00010 = No Error
00011 = Encode
00100 = Low Bit
00101 = High Bit
00110 = Parity
00111 = Two Bus
01000 = Sync
01001 = Gap
01010 = High Word
01011 = Low Word
01100 = No Response
01101 = Reserved Mode
01110 = Late
01111 = Early
10000 = Message Bit Set (Status)
10001 = Reserved Status
10011 = Reserved

WW = Word Type (1553)

00 = Status Word
01 = Data Word
10 = Command Word
11 = RT to RT Command Word

Stack Data (2 bytes) - 1553 Word Value in HEX - MSB-LSB

The SBA will transmit the entire block of records continuously until all requested stack data is transferred (see Section 2 - Step 6). The EOI line will assert on the LRC byte of the last record.

Time Tag

If the stack line is a time tag, find the 24 bit value from the last three bytes of stack data and multiply it by four to equate the time shown on the stack line. No value will be given exceeding 24 bits, but the G bit (fourth) will be set, indicating a value greater than 6.7 E 7.

TIME TAG =

[(Word Description*65535)+(1st Stack Data*256)+(2nd Stack Data)]*4

3.8 GPIB RECORD REVIEW

This section will review record structures sent to the SBA via the GPIB bus. The records descibed in sections 3.2-3.7 are standard SBA records and are common to all GPIB and RS-232 communications. There is, however, a difference in encoding the records depending on the model of SBA you have. SBA-100's and 100A's endcode their records with an DLE-STX-ETX protocol which is common to secure data transmissions. The encoding is shown below.

DLE	STX	SBA RECORD	DLE	ETX	LRC

SBA-100/100A Encoding

The second type of encoding pertains to the SBA-100F/200/200A and has only one byte of encoding, the LRC character. The format appears like the following:

SBA RECORD	LRC

SBA-100F/200/200A Encoding

Both types of encoding will be covered in detail. The first type for SBA-100 and 100A's will be covered in Section 3.8.1 and the second type for SBA-100F and 200 series will be covered in Section 3.8.2. Section 3.8.3 will review the changes that need to be made for SBA 100 or 100A GPIB transmissions to be compatible with SBA-100F and 200 series transmissions.

3.8.1 GPIB Communication Encoding for SBA-100/100A

This type of encoding is common to RS-232 communication equipement and is the type used for all SBA RS-232 transfers (all models). The structure is as follows:

```
-----------------------------------------------
| DLE | STX | SBA RECORD | DLE | ETX | LRC |
-----------------------------------------------
```

DLE = Data Link Escape - 16 - HEX
STX = Start of Text - 02 - HEX
ETX = End of Text - 03 - HEX
LRC = Longitudinal Redundancy Check - Varies with each record.

Besides the identifiers, termination, and LRC characters (standard records) descibed in sections 3.1-3.7, this DLE-STX-ETX sequence must be added to all standard records. This sequence does not alter the structure of a record in any way, but does alter the calculation of the LRC character and may alter the length of a record.

As descibed in section 3.1.1, the LRC character is obtained by exclusive-ORing all the bytes in a record. None of the DLE-STX-ETX characters are included in the calculation EXCEPT the ETX. The ETX must be included in the LRC calculation.

Imbedded DLE's in Records

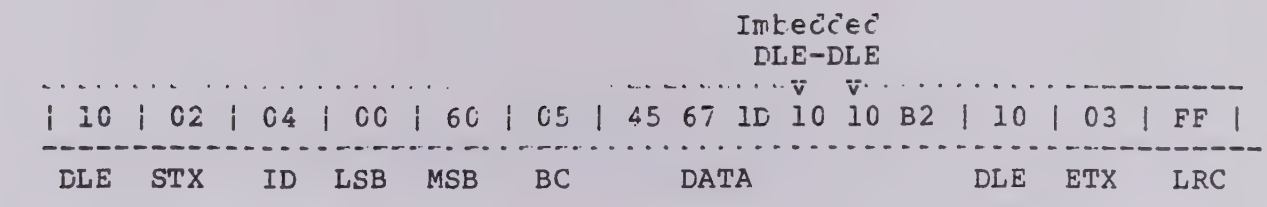
If any data bytes in the SBA RECORD have the same value as the DLE then a second DLE must procede the original DLE. This signals the SBA that the imbedded DLE is part of the SBA RECORD and not the end of the record transmission; an example is shown below.

Within SBA Memory RECORD

```
                                    Imbedded
                                     DLE_DLE
                                       v--v
-------------------------------------------
| 04 | 00 | 60 | 05 | 45 67 1D 10 10 B2 |
-------------------------------------------
  ID  LSB  MSB   BC        DATA
```

See Section 3.1.3 for Memory Record Format

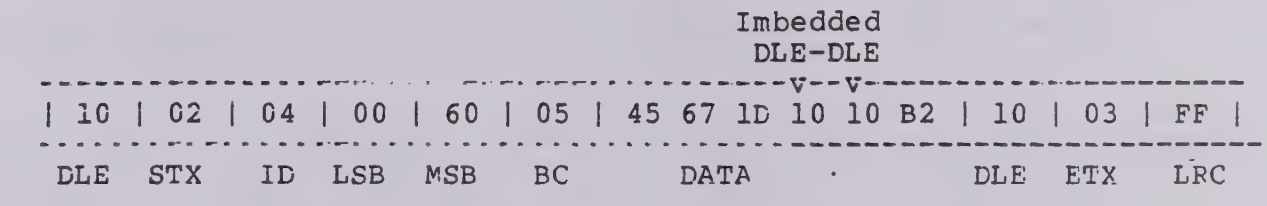
The example above does not show the DLE-STX-ETX-LRC overhead, but it is important to remember that these bytes MUST be included. The following example shows how the entire record would appear (in HEX) when transmitted to the SBA (or from SBA).



The LRC was calculated by exclusive-ORing ALL the bytes between the DLE-STX and DLE-ETX, and ALSO ONLY ONE of the IMBEDDED DLE's and the ENDING ETX.

Record Length

Record length of standard SBA records will change with the addition of the communication encoding described in this section. Most records have a standard length of 256 bytes including the identifier bytes. However, with the addition of the DLE-STX-ETX sequences and the LRC character, the usual maximum length will be 261 bytes. This length may be even greater with the addition of imbedded DLE's. The imbedded DLE's ARE included in the record length, and with an extreme example of a record with nothing but all DLE's, the record length could be 513 bytes. The example below would have a record length of 15 bytes.



The above example should make it apparent that terminating an INPUT or ENTER statement on the length of a record is not a good idea. Terminat the INPUT on the assertion of the EOI line (end of transmission). Your driver program may also want to have a routine to strip out any imbedded DLE's from an SBA record (the example program has an example).

LRC

The best way to calculate the LRC byte in your host computer driver is to have the main routine call a subroutine that calcaulates the LRC, and checks for double DLE's in the output string (skipping one of them in the LRC calculation).

3.8.2 SBA-100F and 200 Series GPIB Encoding

The GPIB encoding for these models consists of only one byte, an Longitudinal Redundancy Check (LRC) character at the end of an SBA RECORD; see below.

This LRC character is simply an exclusive-ORing of all the characters in the SBA RECORD, and signifies the end of a record. For most computers a simple FOR NEXT loop (or equivalent) could be created to exclusive-or the length a string.

This type of overhead is great for transmitting data to the SBA, but could make reading SBA records difficult. When reading multiple record from the SBA (i.e. stack data), it could be hard to determine the end of record if a data byte equaled the LRC value. Thus, the host drive should terminate INPUT or ENTER statements with the assertion of the EOI line, signifiying the end od transmission.

There is no need to imbed DLE's in SBA-100F/200/200A's. The maximum length for an SBA-100F/200/200A record is 257 including the LRC byte.

3.8.3 Comparing GPIB Transmissions Between SBA Models

The GPIB differences between the SBA-100(A), and SBA-100F (and 200 series) can be seen by comparing Sections 3.8.1 , and by reading Section 2.

It should be clear after reading the two previous sections that the difference in the product lines in the removal of the ASCII character sequence (DLE-STX-ETX) from the record protocol. This should not make that big of a change to the host driver's program.

Besides dropping the DLE-STX-ETX encoding scheme, the host driver's program will probably have to make some adjustments as to determining the end of a record transmission by the SBA. Probably the best way is to check the LRC character and compare the ID bytes of records (see Section 3). The best way of course, is to know how long the records ar going to be and only input the respective number of bytes, while comparing the LRC.

Another parameter to check is GOSUB routines that may be changed in ne revisions of software. Call an Application Engineer at Loral for more information when the time comes to check for software addressing changes. These changes should be minor and a little problem to change in the host's programs.

The final difference in the models is that the newer SBA models (100F and 200's) are interrupt driven. This means, that the SBA command ASSIGN CONTROL TO GPIB does not have to be entered, as was (is) required in the older models (100 and 100A). The newer SBA's automatically check for communications from the GPIB port and are constantly in a "ready" state. The older models required the above SB command to force itself into a "ready" state.

US ASCII Character Codes

ASCII Char.	EQUIVALENT FORMS				HP-IB
	Dec	Binary	Oct	Hex	
NUL	0	00000000	000	00	
SOH	1	00000001	001	01	GTL
STX	2	00000010	002	02	
ETX	3	00000011	003	03	
EOT	4	00000100	004	04	SDC
ENQ	5	00000101	005	05	PPC
ACK	6	00000110	006	06	
BEL	7	00000111	007	07	
BS	8	00001000	010	08	GET
HT	9	00001001	011	09	TCT
LF	10	00001010	012	0A	
VT	11	00001011	013	0B	
FF	12	00001100	014	0C	
CR	13	00001101	015	0D	
SO	14	00001110	016	0E	
SI	15	00001111	017	0F	
DLE	16	00010000	020	10	
DC1	17	00010001	021	11	LLO
DC2	18	00010010	022	12	
DC3	19	00010011	023	13	
DC4	20	00010100	024	14	DCL
NAK	21	00010101	025	15	PPU
SYNC	22	00010110	026	16	
ETB	23	00010111	027	17	
CAN	24	00011000	030	18	SPE
EM	25	00011001	031	19	SPD
SUB	26	00011010	032	1A	
ESC	27	00011011	033	1B	
FS	28	00011100	034	1C	
GS	29	00011101	035	1D	
RS	30	00011110	036	1E	
US	31	00011111	037	1F	

ASCII Char.	EQUIVALENT FORMS				HP-IB
	Dec	Binary	Oct	Hex	
space	32	00100000	040	20	LA0
!	33	00100001	041	21	LA1
"	34	00100010	042	22	LA2
#	35	00100011	043	23	LA3
$	36	00100100	044	24	LA4
%	37	00100101	045	25	LA5
&	38	00100110	046	26	LA6
'	39	00100111	047	27	LA7
(	40	00101000	050	28	LA8
)	41	00101001	051	29	LA9
*	42	00101010	052	2A	LA10
+	43	00101011	053	2B	LA11
,	44	00101100	054	2C	LA12
-	45	00101101	055	2D	LA13
.	46	00101110	056	2E	LA14
/	47	00101111	057	2F	LA15
0	48	00110000	060	30	LA16
1	49	00110001	061	31	LA17
2	50	00110010	062	32	LA18
3	51	00110011	063	33	LA19
4	52	00110100	064	34	LA20
5	53	00110101	065	35	LA21
6	54	00110110	066	36	LA22
7	55	00110111	067	37	LA23
8	56	00111000	070	38	LA24
9	57	00111001	071	39	LA25
:	58	00111010	072	3A	LA26
;	59	00111011	073	3B	LA27
<	60	00111100	074	3C	LA28
=	61	00111101	075	3D	LA29
>	62	00111110	076	3E	LA30
?	63	00111111	077	3F	UNL

ASCII Char.	EQUIVALENT FORMS				HP-IB
	Dec	Binary	Oct	Hex	
@	64	01000000	100	40	TA0
A	65	01000001	101	41	TA1
B	66	01000010	102	42	TA2
C	67	01000011	103	43	TA3
D	68	01000100	104	44	TA4
E	69	01000101	105	45	TA5
F	70	01000110	106	46	TA6
G	71	01000111	107	47	TA7
H	72	01001000	110	48	TA8
I	73	01001001	111	49	TA9
J	74	01001010	112	4A	TA10
K	75	01001011	113	4B	TA11
L	76	01001100	114	4C	TA12
M	77	01001101	115	4D	TA13
N	78	01001110	116	4E	TA14
O	79	01001111	117	4F	TA15
P	80	01010000	120	50	TA16
Q	81	1010001	121	51	TA17
R	82	01010010	122	52	TA18
S	83	01010011	123	53	TA19
T	84	01010100	124	54	TA20
U	85	01010101	125	55	TA21
V	86	01010110	126	56	TA22
W	87	01010111	127	57	TA23
X	88	01011000	130	58	TA24
Y	89	01011001	131	59	TA25
Z	90	01011010	132	5A	TA26
[	91	01011011	133	5B	TA27
\	92	01011100	134	5C	TA28
]	93	01011101	135	5D	TA29
^	94	01011110	136	5E	TA30
_	95	01011111	137	5F	UNT

ASCII Char.	EQUIVALENT FORMS				HP-IB
	Dec	Binary	Oct	Hex	
`	96	01100000	140	60	SC0
a	97	01100001	141	61	SC1
b	98	01100010	142	62	SC2
c	99	01100011	143	63	SC3
d	100	01100100	144	64	SC4
e	101	01100101	145	65	SC5
f	102	01100110	146	66	SC6
g	103	01100111	147	67	SC7
h	104	01101000	150	68	SC8
i	105	01101001	151	69	SC9
j	106	01101010	152	6A	SC10
k	107	01101011	153	6B	SC11
l	108	01101100	154	6C	SC12
m	109	01101101	155	6D	SC13
n	110	01101110	156	6E	SC14
o	111	01101111	157	6F	SC15
p	112	01110000	160	70	SC16
q	113	01110001	161	71	SC17
r	114	01110010	162	72	SC18
s	115	01110011	163	73	SC19
t	116	01110100	164	74	SC20
u	117	01110101	165	75	SC21
v	118	01110110	166	76	SC22
w	119	01110111	167	77	SC23
x	120	01111000	170	78	SC24
y	121	01111001	171	79	SC25
z	122	01111010	172	7A	SC26
{	123	01111011	173	7B	SC27
|	124	01111100	174	7C	SC28
}	125	01111101	175	7D	SC29
~	126	01111110	176	7E	SC30
DEL	127	01111111	177	7F	SC31

APPENDIX:
SBA-GPIB SAMPLE PROGRAM

his appendix will present a sample program for the SBA and HP9826 over he GPIB bus. The program covers virtually all aspects of the possible nterface of the SBA with Command, Memory, Header, Lineprinter, and tack record transfers being exercised. The program itself is not ommented, but each page of the program is discussed in this appendix. he paragraphs of this Appendix are numbered according to the line umbers of the program.

lthough the program was written on an 9826 computer, the language hould be simple enough to be used on virtually any computer, with of ourse minor adjustments.

he sections, or routines of this program represent virtually all the ecord transfers that can be performed on an SBA. It is taken for ranted in the program explanation that the reader is familiar with the ecord structures of SBA transfers. The record structures are iscussed in a preceeding module of this Application Note.

urn the page to begin program discussion.

Lines 10-250

These lines are for opening comments and a little fancy intro with a coral header scrolling up the screen. The OUTPUT statement on line 250 is used many times throughout the program to clear the screen by using the characters if Clear$.

Lines 260-690

These lines display the SBA TYPE and GPIB ADDRESS page that questions the user on what type of SBA, the SBA s GPIB address and also checks for entry errors (too many characters). Data is stored in A$ from the INPUT statements. The variable Sba_type is used throughout the program when compare statements must be made for the different SBA GPIB protocols.

```
10    !LORAL INSTRUMENTATION
20    !APPLICATION ENGINEERING
30    !SBA-GPIB PROGRAM
40    !
50    !THIS PROGRAM MAY BE USED FOR ANY SBA
60    !TO TRANSFER COMMAND, MEMORY, HEADER,
70    !LINE PRINTER, AND STACK RECORDS TO AND FROM
80    !ANY SBA.
90    !
100   !  INITIAL PAGE DISPLAY
110   !
120   FOR I=1 TO 19
130     PRINT ""
140   NEXT I
150   PRINT "LORAL INSTRUMENTATION"
160   PRINT "APPLICATION ENGINEERING"
170   PRINT "SBA-GPIB PROGRAM"
180   X=5
190   FOR I=1 TO 15
200     PRINT ""
210   NEXT I
220   BEEP
230   WAIT 2
240   Clear$=CHR$(255)&CHR$(75) !CLEAR SCREEN
250   OUTPUT 2;Clear$;  ! CLEAR CRT
260   !
270   ! SBA TYPE AND GPIB ADDRESS PAGE
280   !
290   PRINT ""
300   PRINT "THIS PROGRAM IS FOR INTERFACING ANY"
310   PRINT "SBA TO THE HP9826 VIA THE GPIB BUS."
320   PRINT ""
330   PRINT "FIRST, THE PROGRAM MUST KNOW WHAT"
340   PRINT "TYPE OF SBA IS BEING USED, AND THE"
350   PRINT "GPIB ADDRESS ON THE SBA'S I/O PAGE."
360   PRINT "MUST PRESS 'ENTER' AFTER EACH ENTRY."
370   PRINT ""
380   PRINT "SBA-100 & 100A'S = 1"
390   PRINT "SBA-100F & 200 & 200A = 0"
400   PRINT ""
410   !
420   INPUT "ENTER 1 OR 0 HERE ",A$
430   IF A$="1" THEN    ! COMPARING SBA TYPES
440     Sba_type=1
450     GOTO 580
460   END IF
470   IF A$="0" THEN
480     Sba_type=0
490   ELSE
500     PRINT ""
510     PRINT "ENTRY ERROR"
520     BEEP
530     WAIT 1
540     GOTO 240
550   END IF
560   !
570   ! READ GPIB ADDRESS
580   !
```

Lines 700-800

Lines 730 and 740 configure the GPIB address for the HP system which requires a 7 in front of the address for I/O port control. Line 750 does the actual assigning of the port. The DIM statements dimension the needed buffers for the program. A$ did not need dimensioning before because the length of the string was small.

Lines 810-1180

Lines 860-1040 clear the screen and display the menu page of the program, giving the user a selection of a function to be performed on the SBA. Lines 1060-1180 do the comparing for the selection chosen and jumps to appropriate subroutine for the function. An entry error section is also included in case the operator made a mistake in selection of function.

```
590   PRINT "ENTER SBA-GPIB ADDRESS."
600   PRINT "MUST BE IN DECIMAL 00-31."
610   !
620   INPUT "ENTER ADDRESS HERE (TWO NUMBERS).",A$
630   IF LEN(A$)>2 THEN
640     PRINT ""
650     PRINT "ENTRY ERROR"
660     BEEP
670     WAIT 1
680     GOTO 240
690   END IF
700   !
710   ! ASSIGNING HP-IB BOARD OF HP.
720   !
730   Address=VAL(A$)
740   Address=Address+700
750   ASSIGN @Sba TO Address  ! ASSIGNING HP I/O
760   !
770   DIM A$[500]
780   DIM Buffer$[500]
790   DIM Buffer1$[500]
800   DIM Temp_buffer$[500]
810   !
820   !
830   !  FUNCTION SELECTION PAGE
840   !
850   !
860   OUTPUT 2;Clear$;
870   IF Sba_type=1 THEN Type$="100 & 100A"
880   IF Sba_type=0 THEN Type$="100F & 200 & 200A"
890   PRINT "GPIB PROGRAM FOR SBA-";Type$
900   PRINT ""
910   PRINT "ENTER THE NUMBER FOR THE DESIRED FUNCTION."
920   PRINT ""
930   PRINT "TO THE SBA                          FROM THE SBA"
940   PRINT "-------------------------------------------"
950   PRINT "1=COMMAND STRING~                   9=STACK READS"
960   PRINT "2=OPERATION MODE SETUPS             0=MEMORY READS"
970   PRINT "3=MEMORY STRINGS"
980   PRINT "4=HEADERS STRINGS"
990   PRINT "5=LINE PRINTER STRINGS"
1000  PRINT ""
1010  PRINT ""
1020  PRINT "AS WITH ALL ENTIES FROM THE HP KEYBOARD,"
1030  PRINT "A CARRIAGE RETURN (ENTER) MUST FOLLOW THE"
1040  PRINT "QUERIED ENTRY."
1050  !
1060  INPUT "ENTER THE NUMBER FOR THE DESIRED FUNCTION.",A$
1070  IF A$="1" THEN GOTO 1190
1080  IF A$="2" THEN GOTO 1620
1090  IF A$="3" THEN GOTO 1620
1100  IF A$="4" THEN GOTO 3140
1110  IF A$="5" THEN GOTO 3560
1120  IF A$="9" THEN GOTO 4030
1130  IF A$="0" THEN GOTO 6270
1140  PRINT ""
1150  PRINT "ENTRY ERROR"
1160  BEEP
```

Lines 1190-1490

These lines display the Command Record function page, allowing the operator to enter a Command string to be sent to the SBA. The string is entered into A$ (line 1400) and assigns the string to Buffer$ which is the main string for all inputs and outputs for the program. An error check is made for strings longer than the specified 256 bytes.

Lines 1500-1610

These lines call a decimal 16 (DLE) string imbedder subroutine for SBA-100. The embedder subroutine also attaches the DLE-STX-ETX sequence to out bound strings. The routine then calls the LRC calculation routine which calculates the LRC for Buffer$.

Line 1560 sets a variable that will be compared in the called routine in line 1570 which outputs Buffer$ to the SBA. All the GPIB address commands are done automatically in an OUTPUT statement in the XMIT routine (line 8220). Lines 1590, 1600, and 1610 provide a break before returning to the beginning of the Command string routine (GOTO 1190).

Lines 1620-1800

These lines compare the function selection for the type of function to be displayed to the operator. Since Operation Mode Setup Records and Memory Records are so similar, the two routines were combined. Thus, the display selection and the setting of the ID byte (Id$) for record transfers are compared and set in these first few lines.

```
1170  WAIT 1
1180  GOTO 830
1190  !
1200  !
1210  !  COMMAND STRING TRANSFER TO SBA
1220  !
1230  !
1240  OUTPUT 2;Clear$;
1250  PRINT "A COMMAND RECORD TRANSFER HAS BEEN SELECTED"
1260  PRINT ""
1270  PRINT "THE STRING TO BE ENTERED CAN BE NO"
1280  PRINT "LONGER THAN 256 ENTRIES. THE FOLLOWING"
1290  PRINT "CHARACTERS WILL AID IN PROGRAMMING THE"
1300  PRINT "SBA THROUGH A COMMAND STRING."
1310  PRINT ""
1320  PRINT "CTRL J = LINE FEED OR DOWN ARROW"
1330  PRINT "CTRL M = CARRIAGE RETURN, OR UP ARROW"
1340  PRINT "SPACE = RIGHT ARROW"
1350  PRINT ""
1360  PRINT "USE ONLY CAPITAL LETTERS"
1370  PRINT ""
1380  PRINT "ENTER Q TO RETURN TO THE MENU."
1390  !
1400  INPUT "ENTER COMMAND STRING HERE.",A$
1410  IF A$="Q" THEN GOTO 830
1420  IF LEN(A$)>256 THEN
1430    PRINT ""
1440    PRINT "ENTRY ERROR"
1450    BEEP
1460    WAIT 1
1470    GOTO 1190
1480  END IF
1490  Buffer$=A$
1500  !
1510  ! CALL DLE CHECK AND LRC CALC - 8000
1520  ! AND XMIT BUFFER$ TO SBA
1530  !
1540  GOSUB 7530
1550  Buffer$=Buffer$&Lrc$
1560  Check_srq=1
1570  GOSUB 8170
1580  OUTPUT 2;Clear$;
1590  PRINT "TRANSFERRING DATA"
1600  WAIT 1
1610  GOTO 1190
1620  !
1630  !
1640  !  OPERATION MODE & MEMORY RECORD TRANSFER
1650  !
1660  !
1670  OUTPUT 2;Clear$;
1680  Bc=0
1690  IF A$="2" THEN
1700    PRINT "OPERATION MODE PAGE UPDATE RECORD"
1710    Id$=CHR$(1)
1720  END IF
1730  !
1740  IF A$="3" THEN
```

Lines 1810-2000

These lines are the executive routine for the selected Operation or Memory function. First of all, these lines set the standard LOAD FROM GPIB SBA command (LG) into Buffer$, call the LRC calculation routine and call the output routine; the LOAD FROM GPIB command puts the SBA into a load state. Then Buffer$ is cleared and the address and data entering routine is called (line 1900).

When the operator has finished entering the address and data to be sent to the SBA (see lines 2150-), the program returns to line 1910 where Buffer$ is redefined. The LRC routines are called again and Buffer$ is temporarily moved to Temp_Buffer$. A zero block record is defined and LRC calculated and Buffer$ is once again redefined for outputting from GOSUB 8220.

Lines 2010-2140

These lines give the transmitting break and displays a message to the operator, explaining that an SBA GOSUB must be performed to setup the SBA system with the loaded Operation Record. The program then returns to the Main Menu.

Lines 2150-2370

This part of the program queries the operator for the beginning address of the record to be transferred to the SBA. The GOSUB 2670 on line 2350 calls a 4 byte ASCII to 2 byte binary conversion routine. The Mem_data variable will be discussed in Lines 2670-3170. Upon the return of the conversion, the address bytes are assigned to the correct position in Buffer$ for outputting.

```
1750    PRINT "MEMORY RECORD TRANSMISSION"
1760    Id$=CHR$(4)
1770  END IF
1780  !
1790  PRINT "HAS BEEN SELECTED."
1800  PRINT ""
1810  !
1820  !  OUTPUT RECORD
1830  !
1840  Buffer$="LG"&CHR$(13)
1850  GOSUB 7530
1860  Buffer$=Buffer$&Lrc$
1870  Check_srq=0
1880  GOSUB 8170
1890  Buffer$=""
1900  GOSUB 2150
1910  Buffer$=Id$&Address_lsb$&Address_msb$&CHR$(Bc)&Buffer$
1920  Check_srq=1
1930  GOSUB 7530
1940  Temp_buffer$=""
1950  Temp_buffer$=Buffer$&Lrc$
1960  Buffer$=""
1970  Buffer$=Id$&CHR$(0)&CHR$(0)&CHR$(0)
1980  GOSUB 7530
1990  Buffer$=Temp_buffer$&Buffer$&Lrc$
2000  GOSUB 8170
2010  !
2020  !
2030  OUTPUT 2;Clear$;
2040  PRINT "TRANSFERRING DATA"
2050  IF Id$=CHR$(1) THEN
2060    PRINT ""
2070    PRINT "THE APPROPRIATE PAGE UP-DATA GOSUB"
2080    PRINT "MUST BE PERFORMED TO SETUP THE SBA."
2090    PRINT "USE A COMMAND STRING TO EXECUTE THE GOSUB."
2100    WAIT 5
2110  END IF
2120  WAIT 1
2130  Check_srq=0
2140  GOTO 810
2150  !
2160  !  READING MEMORY ADDRESS FOR TRANSFERS TO
2170  !  SBA.
2180  !
2190  !  FIRST, READING THE MEMORY ADDRESS
2200  !
2210  PRINT "FIRST, THE PROGRAM MUST KNOW THE"
2220  PRINT "BEGINNING SBA MEMORY ADDRESS FOR"
2230  PRINT "THE DATA TO BE SENT."
2240  PRINT ""
2250  !
2260  INPUT "INPUT THE MEMORY ADDRESS HERE (FOUR DIGIT HEX).",A$
2270  IF A$="Q" THEN GOTO 830
2280  IF LEN(A$)>4 THEN
2290    PRINT "ENTRY ERROR"
2300    BEEP
2310    WAIT 1
2320    GOTO 1620
```

Lines 2380-2660

The first few PRINT statements display a header for entering data and displays the address previously entered. Line 2490 inputs the queried data entry, and is followed by an error length check.

The entry termination character P is check for, and a variable is incremented and printed to keep track of the byte count (BC).

The routine for converting the 2 byte ASCII input to a 1 byte binary output is called (line 2650). When the conversion call has returned, the program jumps back to the original data input line (line 2660).

Lines 2670-3130

This routine converts ASCII bytes into a binary byte(s) for transferring to the SBA. The variable Mem_data determines if the conversion is from 4 byte ASCII to 2 byte binary or 2 byte ASCII to 1 byte binary. The variable C is the shift comparer, the variable X is the positioner for the operator input data, and the variable I is the positioner for assigning the converted data into the correct position of Buffer$.

Lines 2740-2790 subtracts the needed amount to convert a single ASCII HEX entry to a binary HEX variable. The next lines, 2800-2980, shift the first value to the most significant nibble position (Msnib - line 2810), increment the variables, retrieves the next converted byte (line 2850) and add the two numbers, creating a single byte of binary data. The byte is then assigned to the correct position in Buffer$.

The routine then compares for Mem data=1, and if true, then the conversion for memory data is finished and another byte can be retrieved. If Mem_data=0, then the conversion is for the memory address and another binary byte must be converted from two ASCII bytes in the next position of A$. All variables are incremented for counting and assignment purposes.

A final comparison is made to warn the operator that the string in Buffer$ is equal to the specified 256 bytes. If the operator was entering an address, the routine pops back to line 2360. If the operator was entering memory data, the routine pops back to the line 2660 and another byte may be entered.

```
2330  END IF
2340  Mem_data=0
2350  GOSUB 2670
2360  Address_msb$=Buffer$[1,1]
2370  Address_lsb$=Buffer$[2,2]
2380  !
2390  !  DATA ENTRY FROM KEYBOARD FOR MEMORY REC
2400  !
2410  PRINT ""
2420  PRINT "YOU MAY NOW ENTER THE DATA FOR STARTING ADDRESS"
2430  PRINT "                --              --"
2440  PRINT "DATA MUST BE IN TWO DIGIT HEX."
2450  PRINT ""
2460  PRINT "ENTER P WHEN FINISHED ENTERING DATA."
2470  PRINT TABXY(20,10);A$[1,1];A$[2,2];A$[3,3];A$[4,4];
2480  !
2490  INPUT "ENTER DATA HERE (TWO DIGIT HEX).",A$
2500  IF A$="Q" THEN GOTO 830
2510  IF LEN(A$)>2 THEN
2520    PRINT TABXY(0,16);"ENTRY ERROR"
2530    WAIT 1
2540    PRINT TABXY(0,16);"                          "
2550    GOTO 2490
2560  END IF
2570  IF A$="P" THEN
2580    Mem_data=0
2590    RETURN
2600  END IF
2610  Bc=Bc+1
2620  PRINT ""
2630  PRINT TABXY(0,15);"BYTE COUNT =";Bc;"        "
2640  Mem_data=1
2650  GOSUB 2710
2660  GOTO 2490
2670  !
2680  ! CONVERT ASCII TO BINARY
2690  !
2700  I=1
2710  IF Mem_data=1 THEN I=Bc
2720  C=1
2730  X=1
2740  Value=NUM(A$[X,X])
2750  IF Value>57 THEN
2760    Value=Value-55
2770  ELSE
2780    Value=Value-48
2790  END IF
2800  IF C=1 THEN
2810    Msnib=SHIFT(Value,-4)
2820    C=C+1
2830    X=X+1
2840    GOTO 2740
2850  END IF
2860  IF C=2 THEN
2870    Msb=Msnib+Value
2880    Buffer$[I,I]=CHR$(Msb)
2890    I=I+1
2900    C=C+1
```

Lines 3140-3440

The first five program lines clear the CRT, and outputs the LOAD FROM GPIB SBA command to the SBA with the usual LRC calculation and Xmit Buffer$ calls.

The next lines display the Header Record page (PRINT statements) and inputs the queried header string (INPUT) to be sent to the SBA. As usual, an error check is made for entry length.

Lines 3450-3550

These lines assign Buffer$ with the input string and proper header byte protocol (line 3500), calls the LRC converter (line 3510), reassigns Buffer$, and calls the XMIT Buffer$ routine (line 3540). Then the usual program delay and message is displayed and the program returns to the Main Menu.

```
2910    IF Mem_data=0 THEN
2920      X=X+1
2930    ELSE
2940    END IF
2950    GOTO 2740
2960  END IF
2970  IF Mem_data=1 THEN GOTO 3090
2980  IF C=3 THEN
2990    Lsnib=SHIFT(Value,-4)
3000    C=C+1
3010    X=X+1
3020    GOTO 2740
3030  END IF
3040  IF C=4 THEN
3050    Lsb=Lsnib+Value
3060    Buffer$[I,I]=CHR$(Lsb)
3070    I=I+1
3080  END IF
3090  !
3100  IF LEN(Buffer$)>256 THEN
3110    PRINT "MAX STRING LENGTH"
3120  END IF
3130  RETURN
3140  !
3150  !
3160  !  HEADER RECORD TRANSFERS
3170  !
3180  !
3190  OUTPUT 2;Clear$;
3200  !
3210  Buffer$="LG"&CHR$(13)
3220  GOSUB 7530
3230  Buffer$=Buffer$&Lrc$
3240  GOSUB 8170
3250  !
3260  PRINT "HEADER TRANSFERS HAS BEEN CHOSEN"
3270  PRINT ""
3280  PRINT ""
3290  PRINT "ENTER ANY ASCII CHARACTER BELOW FOR"
3300  PRINT "DISPLAY ON THE SBA'S COMMAND LINE."
3310  PRINT "THE ASCIII STRING CAN BE NO LONGER"
3320  PRINT "THAN 37 CHRACTERS."
3330  PRINT ""
3340  PRINT "ENTER Q TO RETURN TO THE MENU."
3350  !
3360  INPUT "ENTER STRING HERE.",A$
3370  IF A$="Q" THEN GOTO 830
3380  IF LEN(A$)>37 THEN
3390    PRINT ""
3400    PRINT "ENTRY ERROR"
3410    BEEP
3420    WAIT 1
3430    GOTO 3140
3440  END IF
3450  !
3460  Buffer$=CHR$(3)&A$&CHR$(3)
3470  GOSUB 7530
3480  Buffer$=Buffer$&Lrc$
```

Lines 3560-4020

These lines are almost identical to those described on the preceeding page, except the string length can be longer, and the protocol assigned to Buffer$ is different (line 3970).

```
3490  Check_srq=1
3500  GOSUB 8170
3510  Check_srq=0
3520  OUTPUT 2;Clear$;
3530  PRINT "TRANSFERRING DATA"
3540  WAIT 1
3550  GOTO 810
3560  !
3570  !
3580  !  LINE PRINTER RECORD TRANSFERS
3590  !
3600  !
3610  OUTPUT 2;Clear$;
3620  !
3630  Buffer$="LG"&CHR$(13)
3640  GOSUB 7530
3650  Buffer$=Buffer$&Lrc$
3660  GOSUB 8170
3670  !
3680  PRINT "LINE PRINTER RECORD TRANSFER"
3690  PRINT "HAS BEEN CHOSEN."
3700  PRINT ""
3710  PRINT ""
3720  PRINT "ENTER BELOW THE ASCII STRING TO BE OUTPUT"
3730  PRINT "TO THE ASSIGNED SBA LINE PRINTER PORT."
3740  PRINT "THE STRING CAN BE NO LONGER THAN 256"
3750  PRINT "CHRACTERS. THE FOLLOWING CHARACTERS"
3760  PRINT "MAY HELP IN ENCODING THE STRING."
3770  PRINT ""
3780  PRINT "CTRL M = CARRIAGE RETURN"
3790  PRINT "CTRL J = LINE FEED"
3800  PRINT ""
3810  PRINT "ENTER Q TO RETURN TO THE MENU."
3820  !
3830  INPUT "ENTER ASCII STRING HERE.",A$
3840  IF A$="Q" THEN GOTO 830
3850  IF LEN(A$)>256 THEN
3860    PRINT ""
3870    PRINT "ENTRY ERROR"
3880    BEEP
3890    WAIT 1
3900    GOTO 3560
3910  END IF
3920  !
3930  Buffer$=CHR$(2)&"XX"&A$&CHR$(6)
3940  GOSUB 7530
3950  Buffer$=Buffer$&Lrc$
3960  Check_srq=1
3970  GOSUB 8170
3980  Check_srq=0
3990  OUTPUT 2;Clear$;
4000  PRINT "TRANSFERRING DATA"
4010  WAIT 1
4020  GOTO 810
4030  !
4040  !
4050  !  READING STACK DATA FROM THE SBA
4060  !
```

Lines 4030-4430

These lines display the address quering and initial Stack Record displays. Lines 4080-4200 display the page, asking for the beginning and ending address, and lines 4220-4430 input the addresses, error check the length and reassign and addresses the alternate string variables.

Lines 4440-4550

Lines 4450 and 4460 assign the correct values to Buffers$ (SAVE STACK XXXX XXXX to GPIB - SBA command where X represents Begin$ and End$) and call the LRC calculation routine. The HP screen is cleared and a wait statement is displayed while Buffer$ is transmitted. The program then enters the data that the SBA transmitted (Stack data) from the command sent. The ENTER statement on line 4520 is assigned special characteristics for entering data until the EOI line is detected on the GPIB bus (-K,%) without regard to terminating characters (such as line feeds). The entered data in Buufer$ is then stripped of embedded DLE characters (if operating an SBA-100 or 100A), and a serial poll is conducted on the SBA by using part of the XMIT Buffer$ routine.

Lines 4560-4690

Lines 4590-4630 print the Stack data display page headers with the address range that the operator entered displayed. Lines 4650-4690 determine the starting position for reading the encoded stack data. When decoding stack data, the programmer does not need to look at the first four bytes (six bytes with SBA-100 s and 100A s because of the beginning DLE and STX characters) of stack records transmitted to a host computer because they are exactly the same for each record. Thus, a variable is set to the correct value which will be used to extract bytes out of Buffer$. Lines 4650-4690 set the variable according to the type of SBA being used.

```
4070  !
4080  OUTPUT 2;Clear$;
4090  PRINT "STACK READ HAS BEEN SELECTED."
4100  PRINT ""
4110  PRINT ""
4120  PRINT "WHAT BEGINNING AND ENDING ADDRESS DO YOU"
4130  PRINT "WANT READ? ENTER THE MOST SIGNIFICANT"
4140  PRINT "ADDRESS FIRST (FOUR DECIMAL ENTRIES)."
4150  PRINT ""
4160  PRINT "THE MAXIMUM DISPLAY IS 10 LINES AT A TIME."
4170  PRINT ""
4180  PRINT "PROGRAM WILL NOT RUN IF THE STACK HAS NOT"
4190  PRINT "BEEN INITIALIZED (RUN STACK)."
4200  PRINT TABXY(0,24);"ENTER Q TO RETURN TO MENU."
4210  !
4220  INPUT "ENTER THE BEGINNING ADDRESS HERE (FOUR-DEC).",A$
4230  IF A$="Q" THEN GOTO 330
4240  IF LEN(A$)<4 THEN
4250    PRINT ""
4260    PRINT "ENTRY ERROR"
4270    BEEP
4280    WAIT 1
4290    GOTO 3320
4300  END IF
4310  Begin$=A$
4320  PRINT TABXY(42,24);Begin$
4330  !
4340  INPUT "ENTER ENDING ADDRESS HERE (FOUR-DEC).",A$
4350  IF A$="Q" THEN GOTO 10000
4360  IF LEN(A$)<4 THEN
4370    PRINT ""
4380    PRINT "ENTRY ERROR"
4390    BEEP
4400    WAIT 1
4410    GOTO 3320
4420  END IF
4430  End$=A$
4440  !
4450  Buffer$="S3"&Begin$&End$&"G"&CHR$(13)
4460  GOSUB 7530
4470  Buffer$=Buffer$&Lrc$
4480  OUTPUT 2;Clear$;
4490  PRINT "PLEASE WAIT"
4500  GOSUB 3170
4510  Buffer$=""
4520  ENTER @Sba USING "-K,%";Buffer$
4530  IF Sba_type=1 THEN GOSUB 7980
4540  Check_srq=1
4550  GOSUB 3210
4560  !
4570  !  CONVERTING STACK DATA
4580  !
4590  OUTPUT 2;Clear$;
4600  PRINT "STACK DATA RECEIVED. DATA DISPLAYED FOR"
4610  PRINT "ADDRESSES ";Begin$;" TO ";End$;"."
4620  PRINT ""
4630  PRINT TABXY(0,24);"ENTER Q TO RETURN TO MENU."
4640  !
```

Lines 4720-4790

The first two lines (4720,4730) extract the LSB-MSB stack address configuration out of Buffer$, converting the string characters to number values and adding the two together to arrive at a printable single byte value.

Line 4740 also converts the third string character of the six encoded stack bytes to a number value which can be used for BIT comparisons. Lines 4750-4790 determine, through the use of BIT value extractions, if the word value associated with this record was a PRImary bus word or SECondary bus word (1553- from stack lines).

Lines 4800-5060

Lines 4810 and 4820 clear the unused bits out of Word_desc1 for the error type comparisons in lines 4830-4990. If a certain value is detected in the five bits of Word_desc2, then the appropriate error needs to be assigned to a variable for display with the word sent over in this respective Stack record. The error type is stored in Err_type$ for printing later in the routine.

Lines 5000-5060 are for determining the type of 1553 word sent over in the Stack record. Line 5030 is a repeat of line 4740 and really is not necessary. Lines 5010 and 5020 again clears out the unsed bits for the value comparison in lines 5030-5060 (1553 word type).

There are many different ways to clear the unused bits in a word, such as binary logic operators, masking, etc., for it is neccessary for these bits to be cleared since the different fields of this byte are completely independent of each other.

Lines 5080-5130

These lines go back to byte two of the stack record data to see if the encoded stack word of the record was a new command detected by the SBA. The outcome of a bit comparison is assigned to a string variable for future printing.

Lines 5140-5240

Line 5150 determines if the stack record data is a time tag, and if so, then preform the time tag conversion on the next three bytes of the stack record data. The three bytes value is really a 24 bit equivelent and the proper multiplication conversions must be made (lines 5160-5200 - bit shifting could have been used). If bit 3 of the Rec_type variable (line 5080) is set then the time tag is greater than 6.5 E 7 and the variable is assigned. A jump is then made to avoid any more needless word conversions since the stack record represents a time tag. This comparison should have been done earlier to avoid the needless bit comparisons already made. Stupid me.

```
4650  IF Sba_type=1 THEN
4660    P=7
4670  ELSE
4680    P=5
4690  END IF
4700  Y=6
4710  X=0
4720  Addr_byte2=NUM(Buffer$[P+1,P+1])
4730  Addr_byte=ROTATE(Addr_byte2,-8)+NUM(Buffer$[P,P])
4740  Word_desc1=NUM(Buffer$[P+3,P+3])
4750  IF BIT(Word_desc1,7)=1 THEN
4760    Bus_type$="PRI"
4770  ELSE
4780    Bus_type$="SEC"
4790  END IF
4800  !
4810  Word_desc2=SHIFT(Word_desc1,-9)
4820  Word_desc2=SHIFT(Word_desc2,11)
4830  IF Word_desc2=2 THEN Err_type$="      "
4840  IF Word_desc2=3 THEN Err_type$="ENCODE"
4850  IF Word_desc2=4 THEN Err_type$="LOWBIT"
4860  IF Word_desc2=5 THEN Err_type$="HIBIT "
4870  IF Word_desc2=6 THEN Err_type$="PARITY"
4880  IF Word_desc2=7 THEN Err_type$="TWOBUS"
4890  IF Word_desc2=8 THEN Err_type$="SYNC  "
4900  IF Word_desc2=9 THEN Err_type$="GAP   "
4910  IF Word_desc2=10 THEN Err_type$="HI WRD"
4920  IF Word_desc2=11 THEN Err_type$="LOWWRD"
4930  IF Word_desc2=12 THEN Err_type$="NORESP"
4940  IF Word_desc2=13 THEN Err_type$="RMODE "
4950  IF Word_desc2=14 THEN Err_type$="LATE  "
4960  IF Word_desc2=15 THEN Err_type$="EARLY "
4970  IF Word_desc2=16 THEN Err_type$="MEBIT "
4980  IF Word_desc2=17 THEN Err_type$="RSTAT "
4990  IF Word_desc2=18 THEN Err_type$="RESERV"
5000  Word_desc1=NUM(Buffer$[P+3,P+3])
5010  Word_desc2=SHIFT(Word_desc1,-14)
5020  Word_desc2=SHIFT(Word_desc2,8)
5030  IF Word_desc2=0 THEN Word_type$="S"
5040  IF Word_desc2=64 THEN Word_type$="D"
5050  IF Word_desc2=128 THEN Word_type$="C"
5060  IF Word_desc2=192 THEN Word_type$="R"
5070  !
5080  Rec_type=NUM(Buffer$[P+2,P+2])
5090  IF BIT(Rec_type,0)=1 THEN
5100    New_command$="N"
5110  ELSE
5120    New_command$=" "
5130  END IF
5140   !
5150  IF BIT(Rec_type,1)=1 THEN
5160    Tt=1
5170    Time_tag=NUM(Buffer$[P+3,P+3])*65536
5180    Time_tag=(Time_tag)+(NUM(Buffer$[P+4,P+4])*256)
5190    Time_tag=((Time_tag)+(NUM(Buffer$[P+5,P+5])))*4
5200    IF BIT(Rec_type,3)=0 THEN
5210      Time_tag=65100000
5220    END IF
```

Lines 5250-5340

These lines get the 1553 data word embedded in a stack record. The first half of the 1553 word is extrated and rotated from Buffer$, and then added to the second half of the 1553 word extracted from Buffer$. The word type is then determined because there are different printing formats for data words than command or status words. The appropriate GOSUB is called to encode the word for printing on the HP screen.

Lines 5350-5560

These lines perform the printing on the HP screen. The first seven lines print the values for which all the bit comparisons were made earlier. Lines 5470-5560 check to see whether a data word or command/status word should be printed (the string values were calculated in later routines). It also checks to see if a time tag is to be printed instead of a word from which a GOTO jumped earlier.

Lines 5570-5690

These lines increment the position in Buffer$ to the next stack record according to the type of SBA being used. Also, a determination is made to see if the positioner (P) in Buffer$ has reached the end of transmitted stack block. If the end of the block has been reached, then the program jumps back to the addressing quering of this routine.

```
5230    GOTO 5520
5240  END IF
5250  !
5260  Byte_data=ROTATE(NUM(Buffer$[P+4,P+4]),-8)+NUM(Buffer$[P+5,P+5])
5270  IF Word_type$<>"D" THEN
5280    Hex1=0
5290    Hex2=0
5300    Hex3=0
5310    GOSUB 5700
5320  ELSE
5330    GOSUB 7170
5340  END IF
5350  !
5360  !  ACTUAL PRINTING
5370  !
5380  PRINT TABXY(0,4)," ADDR BUS  ERROR      WD VALUE"
5390  PRINT TABXY(0,5),"-----------------------------------"
5400  PRINT TABXY(0,Y);Addr_byte
5410  PRINT TABXY(7,Y);Bus_type$
5420  PRINT TABXY(12,Y);Err_type$
5430  PRINT TABXY(23,Y);Word_type$
5440  PRINT TABXY(22,Y);New_command$
5450  !
5460  !
5470  IF Word_type$<>"D" THEN
5480    PRINT TABXY(24,Y);Cmd_stat$
5490  ELSE
5500    PRINT TABXY(24,Y);Data_byte$
5510  END IF
5520  IF Tt=1 THEN
5530    PRINT TABXY(0,Y);Addr_byte
5540    PRINT TABXY(25,Y);Time_tag;" USEC"
5550    Tt=0
5560  END IF
5570  !
5580  !  INCREMENT POSITIONERS AND CHECK LENGTH
5590  !
5600  Y=Y+1
5610  IF Sba_type=1 THEN
5620    P=P+15
5630  ELSE
5640    P=P+11
5650  END IF
5660  IF P+3>LEN(Buffer$) THEN
5670    GOTO 4220
5680  END IF
5690  GOTO 4720
5700  !
5710  ! STACK COMMAND AND STATUS WORD DISPLAY
5720  ! ROUTINES.
5730  !
5740  Ms_nib=0
5750  Ls_nib=0
5760  S1=-1
5770  Sr=12
5780  F=15
5790  GOSUB 6070
5800  Hex1$=Ms_nib$&Ls_nib$
```

Lines 5700-6260

This subroutine performs the conversion for printing the binary stack data for command or status words in a hex format. There is four parts of the routine, each of which have the correct shift left (Sl) and shift right (Sr) variables declared; the shifting is for clearing the unused bits. The second part of the routine does not need the shifting, because the conversion is performed on only one bit. The other parts of the routine (lines 5740-6040) call a value manipulator and hex conversion sub-subroutine where the actual shifting and comparisons are made. The final value for the command or status word is stored in Cmd_stat$ for printing.

Lines 6270-6530

These lines display the Memory read record page and queries the operator for the beginning and ending addresses to be read from the SBA. The usual error checks are made after each INPUT and the respective addresses are reassigned to alternate strings.

```
5810  !
5820  IF Word_type$="C" THEN
5830    IF BIT(Byte_data,10)=1 THEN
5840      Tr$="T"
5850    ELSE
5860      Tr$="R"
5870    END IF
5880  ELSE
5890    IF BIT(Byte_data,10)=1 THEN
5900      Tr$="1"
5910    ELSE
5920      Tr$="0"
5930    END IF
5940  END IF
5950  S1=-7
5960  F=9
5970  GOSUB 6050
5980  Hex2$=Ms_nib$&Ls_nib$
5990  S1=-12
6000  F=4
6010  GOSUB 6050
6020  Hex3$=Ms_nib$&Ls_nib$
6030  Cmd_stat$=Hex1$&" "&Tr$&" "&Hex2$&" "&Hex3$
6040  RETURN
6050  !
6060  ! DEC TO STACK HEX CONVERSION
6070  !
6080  IF BIT(Byte_data,F)=1 THEN
6090    Ms_nib$="1"
6100  ELSE
6110    Ms_nib$="0"
6120  END IF
6130  Ls_nib=SHIFT(Byte_data,S1)
6140  Ls_nib=SHIFT(Ls_nib,Sr)
6150  IF Ls_nib>9 THEN
6160    IF Ls_nib=10 THEN Ls_nib$="A"
6170    IF Ls_nib=11 THEN Ls_nib$="B"
6180    IF Ls_nib=12 THEN Ls_nib$="C"
6190    IF Ls_nib=13 THEN Ls_nib$="D"
6200    IF Ls_nib=14 THEN Ls_nib$="E"
6210    IF Ls_nib=15 THEN Ls_nib$="F"
6220  ELSE
6230    Ls_nib=Ls_nib+48
6240    Ls_nib$=CHR$(Ls_nib)
6250  END IF
6260  RETURN
6270  !
6280  !
6290  !   MEMORY READ FROM SBA
6300  !
6310  !
6320  OUTPUT 2;Clear$;
6330  PRINT "A MEMORY READ FROM THE SBA HAS BEEN CHOSEN."
6340  PRINT ""
6350  PRINT "FIRST, THE BEGINNING AND ENDING ADDRESS"
6360  PRINT "MUST BE ENTERED.  THE MAXIMUM DISPLAY"
6370  PRINT "OF THIS PROGRAM IS 35 BYTES OF DATA, SO"
6380  PRINT "DO NOT ENTER AN ADDRESS RANGE GREATER THAN"
```

Lines 6650-6810

These lines, as you should be able to see by now, assign the Memory record protocols and addresses to Buffer$. The lines call out the LRC calculation, and XMIT routines while displaying a wait message to the operator. The requested data from the SBA is then ENTERed (EOI termination only -K,%), the DLE stripper is called for SBA-100 and 100A models, and the serial poll portion of the XMIT routine is called.

Lines 6820-6910

The first five program lines (6830-6870) set the positioner (dependent on the SBA model) for extracting the memory address from Buffer$. The address is converted from two byte binary to one byte binary (6880) and a subroutine is called out to convert the single byte to a four digit hex string byte for printing purposes.

```
6390  PRINT "32 BYTES."
6400  PRINT ""
6410  PRINT TABXY(0,23),"ENTER Q TO RETURN TO MENU."
6420  !
6430  INPUT "ENTER THE BEGINNING ADDRESS (4 HEX ENTRIES).",A$
6440  IF A$="Q" THEN GOTO 810
6450  IF LEN(A$)<4 THEN
6460    PRINT ""
6470    PRINT "ENTRY ERROR"
6480    BEEP
6490    WAIT 1
6500    GOTO 6320
6510  END IF
6520  Begin_addr$=A$
6530  PRINT TABXY(42,23),Begin_addr$
6540  !
6550  INPUT "ENTER THE ENDING ADDRESS (4 HEX ENTRIES).",A$
6560  IF A$="Q" THEN GOTO 810
6570  IF LEN(A$)<>4 THEN
6580    PRINT ""
6590    PRINT "ENTRY ERROR"
6600    BEEP
6610    WAIT 1
6620    GOTO 6320
6630  END IF
6640  End_addr$=A$
6650  !
6660  Buffer$=""
6670  Buffer$="3M"&Begin_addr$&End_addr$&"G"&CHR$(13)
6680  GOSUB 7530
6690  Buffer$=Buffer$&Lrc$
6700  OUTPUT 2;Clear$;
6710  PRINT "PLEASE WAIT"
6720  GOSUB 8170
6730  Buffer$=""
6740  ENTER @Sba USING "-K,%";Buffer$
6750  IF Sba_type=1 THEN GOSUB 7980
6760  Check_srq=1
6770  GOSUB 3210
6780  !
6790  OUTPUT 2;Clear$;
6800  PRINT "MEMORY DATA RECEIVED FOR BEGINNING"
6810  PRINT "ADDRESS ";Begin_addr$;"."
6820  !
6830  IF Sba_type=1 THEN
6840    P=4
6850  ELSE
6860    P=2
6870  END IF
6880  Byte_data=SHIFT(NUM(Buffer$[P+1,P+1]),-8)+NUM(Buffer$[P,P])
6890  GOSUB 7170
6900  Addr_byte$=""
6910  Addr_byte$=Data_byte2$
6920  !
6930  X=8
6940  Y=4
6950  P=P+2
6960  Pc=1
```

Lines 6920-7160

The first few lines set the pointers used to print the memory data sent from the SBA and stored in Buffer$. First, the beginning address is printed (6980); only five memory words are printed per line on the HP screen, and only one address per five words.

The FOR-NEXT loop performs the following: extracts the respective data word (according to the pointer -7000); calls the one byte binary to four byte hex conversion routine (memory words are only 8-bits, thus, the first two digits will always be zero); increments the printing count (Pc); compares Pc for starting a new line and printing the address for the new line. If a new line is to be printed, one is added to the old memory address value, the hex conversion routine is called, the pointers are set or reset, the memory address is printed and a jump is performed to the NEXT line (7150). If a new line is not to be printed, then the printing pointer (X) is incremented for printing memory data.

Lines 7170-7520

This routine performs shifting in increments of nibbles to a 16-bit binary word, and converts the isolated nibbles into a HEX character. This routine is used for stack data and memory addresses and data from the SBA.

```
6970  Addr_incr=5
6980  PRINT TAB(Y(0,4),Begin_addr$;" - "
6990  FOR I=1 TO NUM(Buffer$[P,P])
7000    Byte_data=NUM(Buffer$[P+I,P+I])
7010    GOSUB 7170
7020    PRINT TABXY(X,Y),Data_byte$
7030    Pc=Pc+1
7040    IF Pc>5 THEN
7050      Byte_data=SHIFT(NUM(Buffer$[P-1,P-1]),-8)+NUM(Buffer$[P-2,P-2])+Addr_i
7060      GOSUB 7190
7070      Pc=1
7080      Addr_incr=Addr_incr+5
7090      X=8
7100      Y=Y+2
7110      PRINT TABXY(0,Y),Data_byte1$;Data_byte2$;" - "
7120      GOTO 7150
7130    END IF
7140    X=X+8
7150  NEXT I
7160  GOTO 6410
7170  !
7180  !
7190  !  DATA WORD HEX CONVERSION FOR STACK
7200  !  MEMORY DISPLAYS.
7210  !
7220  !
7230  Hex=SHIFT(Byte_data,12)
7240  GOSUB 7410
7250  Data_byte1$=Hex$
7260  Hex=SHIFT(Byte_data,-4)
7270  Hex=SHIFT(Hex,12)
7280  GOSUB 7410
7290  Data_byte1$=Data_byte1$&Hex$
7300  Hex=SHIFT(Byte_data,-8)
7310  Hex=SHIFT(Hex,12)
7320  GOSUB 7410
7330  Data_byte2$=Hex$
7340  Hex=SHIFT(Byte_data,-12)
7350  Hex=SHIFT(Hex,12)
7360  GOSUB 7410
7370  Data_byte2$=Data_byte2$&Hex$
7380  Data_byte$=Data_byte1$&Data_byte2$
7390  RETURN
7400  !
7410  IF Hex<9 THEN
7420    IF Hex=10 THEN Hex$="A"
7430    IF Hex=11 THEN Hex$="B"
7440    IF Hex=12 THEN Hex$="C"
7450    IF Hex=13 THEN Hex$="D"
7460    IF Hex=14 THEN Hex$="E"
7470    IF Hex=15 THEN Hex$="F"
7480  ELSE
7490    Hex=Hex+48
7500    Hex$=CHR$(Hex)
7510  END IF
7520  RETURN
7530  !
7540  !
```

Lines 7530-7780

These lines are the first routine of three which insert DLE s (decimal 16), calculate the LRC characters on Buffer$ for outgoing data, and strip embedded DLE characters from ENTERed data. Lines 7610-7780 checks to see if the outgoing data in Buffer$ (by the use of a FOR-NEXT) contains a DLE character (7670), and if it does, adds another DLE character in a temporary Buffer1$ (7690). The temporary Buffer1$ is then reassigned the Buffer$. Another check is performed to see if the DLE-STX-ETX sequence needs to be added to Buffer$ (which it does for SBA-100's and 100A's). The routine then calls the LRC calculation routine directly below.

Lines 7790-7950

This LRC calculation routine loops to the length of Buffer$ and exculsive-ors the bytes against one another (7890). The loop checks for single and double DLE characters (7870) and simply increments over the first or only DLE character (7940) and calculates the LRC on the next ensuing byte. The final LRC value is converted to a string character and stored in LRC$.

Lines 7960-8140

This routine loops to the length of Buffer$, checking for a double DLE sequence (8010 and 8020). If the double compare is found true, then the positioner for Buffer$ is incremented and only one of the DLE characters (the last one) is moved over to Buffer1$. If only one DLE character is found then Buffer1$=Buffer$, and the same is true if the DLE compare is false.

```
7550  ! RECORD PROTOCOL ROUTINES:LRC, ADDING AND
7560  ! DELETING DLE'S.
7570  !
7580  !  ADDING DLE'S TO A STRING
7590  !
7600  !
7610  H=1
7620  Y=0
7630  IF Sba_type=0 THEN GOTO 7790
7640  Buffer1$=Buffer$
7650  FOR I=1 TO LEN(Buffer$)
7660    Buffer1$[H,H]=Buffer$[I,I]
7670    IF Buffer$[I,I]=CHR$(16) THEN
7680      H=H+1
7690      Buffer1$[H,H]=CHR$(16)
7700    END IF
7710    H=H+1
7720  NEXT I
7730  Buffer$=Buffer1$
7740  IF Sba_type=1 THEN
7750    Buffer$=CHR$(16)&CHR$(2)&Buffer$&CHR$(16)&CHR$(3)
7760  END IF
7770  GOSUB 7790
7780  RETURN
7790  !
7800  !  LRC CALCULATION ON BUFFER$
7810  !
7820  C=1
7830  Numb=0
7840  Lrc=0
7850  IF Sba_type=1 THEN C=3
7860  FOR I=C TO LEN(Buffer$)
7870    IF Sba_type=1 THEN
7880      IF Buffer$[I,I]=CHR$(16) THEN GOSUB 7960
7890    END IF
7900    Numb=NUM'Buffer$[I,I])
7910    Lrc=BINEOR(Numb,Lrc)
7920  NEXT I
7930  Lrc$=CHR$(Lrc)
7940  RETURN
7950  !
7960  I=I+1
7970  RETURN
7980  !
7990  !  IMBEDDED DLE STRIPPER FOR INCOMING DATA
8000  !
8010  H=1
8020  FOR I=1 TO LEN(Buffer$)
8030    IF Buffer$[I,I]=CHR$(16) THEN
8040      IF Buffer$[I+1,I+1]=CHR$(16) THEN
8050        I=I+1
8060        Buffer1$[H,H]=Buffer$[I,I]
8070      END IF
8080      Buffer1$[H,H]=Buffer$[I,I]
8090      H=H+1
8100    ELSE
8110      Buffer1$[H,H]=Buffer$[I,I]
8120      H=H+1
```

Lines 8150-8310

These lines provide the XMIT Buffer$ routine (8180). A variable is checked also to see if a serial poll should be conducted on the SBA; the variable is set independently from other routines. The compare on the input serial poll status byte (8210 and 8220) only checks for bad LRC's transmitted to the SBA. The program could be configured to check all the possible returned values from the SBA. The values are found earlier in this Application Note.

The OUTPUT and SPOLL statements automatically perform the GPIB command assignments to the SBA (such as addressing).

Finally, this sample program is ended on line 8310.

NOTE: This program was developed quickly and as usual, between many other more pressing projects. While typing out this review of the program, many small programming adjustments should have been made, but time would not permit. It was the goal of this program to provide the user with clear methods of decoding and encoding SBA data. Good luck with your program and if there are any questions, please call an Application Engineer at 1-800-541-0419 (in Calif. 619-560-5888 ext. 4604).

Richard Schuh, 2-86

```
8130    END IF
8140  NEXT I
8150  Buffer$=Buffer1$
8160  RETURN
8170  !
8180  !   XMIT BUFFER$ TO SBA
8190  !
8200  OUTPUT @Sba;Buffer$;END
8210  WAIT .5
8220  IF Check_srq=1 THEN
8230    Srq=SPOLL(Address) !  SERIAL POLL
8240    IF Srq=193 THEN
8250      PRINT ""
8260      PRINT "LRC ERROR - RESTART PROGRAM"
8270      BEEP
8280      WAIT 2
8290    END IF
8300  END IF
8310  RETURN
8320  !
8330  END
```

4.5. DIRECT PAGE MEMORY ADDRESSES

Direct Page Programming (Memory Address) is the method by which you use the memory address of a variable to program the parameter value you want. This section defines the variable addresses of each page, and describes the legal contents of each variable.

OPER MODE 1

```
                                        03:58:28
                                        HEX 1553
   DISPLAY OPER MODE 1                   PAGE 00

   OPERATION MODE:
NORESP (A=10,B=12,C=5,D=17,E=50)US--B  ----------4A9F   NOTE 1
BROADCAST COMMANDS ----------NO  ----------4AA0   NOTE 2
BUS SELECT----------INT  ----------4AA1   NOTE 3
OUTPUT AMPLITUDE (COUNT)----------128  ----------4AA2   NOTE 4
XMIT FREQ----------NOR  ----------4AA5   NOTE 5
ZERO XING DEVIATION (NSEC)----------A, +150  ----------4AA6   NOTE 6

   FUNCTION DEFINITION
FUNCTION----------RAND  ----------4AA7   NOTE 7
UPPER LIMIT----------00000  ----------4AA8   NOTE 8
LOWER LIMIT----------00000  ----------4AAD   NOTE 9
CYCLE----------001  ----------4AB2   NOTE 10
STEP----------001  ----------4AB5   NOTE 11
```

NOTE 1: A=31H,B=30H,C=32H,D=33H,E=34H

NOTE 2: NO=30H,YES=31H

NOTE 3: INT=32H,DIR=31H,XFM=30H

NOTE 4: THREE BYTE ASCII HEX FIELD. 4AA2=MSB 4AA4=LSB

NOTE 5: NOR=30H,A(+100)=31H,B(-100)=32H,C(+1K)+33H,D(-1K)=34H

NOTE 6: A,< +150=30H B,> -150=31H C, > +150=32H D, < -150=33H

NOTE 7: RAND=30H,SAW=31H

NOTE 8: FIVE BYTE ASCII FIELD. 4AA8=MSB 4AAC=LSB

NOTE 9: FIVE BYTE ASCII FIELD. 4AAD=MSB 4AB1=LSB

NOTE 10: THREE BYTE ASCII FIELD. 4AB2=MSB 4AB4=LSB

NOTE 11: THREE BYTE ASCII FIELD. 4AB5=MSB 4AB7=LSB

OPER
MODE 2

04:04:12
HEX 1553
DISPLAY OPER MODE 2 PAGE 01

RESERVED STATUS BIT--1111111111 ----------4BBE NOTE 1

	T/R	MCODE	RSVD*	T/R	MCODE	RSVD*	T/R	MCODE	RSVD*	COL1 ADD	COL2 ADD	COL3 ADD	
1	00000	NO	*	1	00001	NO	* 1	00010	NO	4BCE	4BD5	4BDC	NOTE 2
1	00011	NO	*	1	00100	NO	* 1	00101	NO	4BE3	4BEA	4BF1	NOTE 2
1	00110	NO	*	1	00111	NO	* 1	01000	NO	4BF8	4BFF	4C06	NOTE 2
1	01001	YES	*	1	01010	YES	* 1	01011	YES	4C0D	4C14	4C18	NOTE 2
1	01100	YES	*	1	01101	YES	* 1	01110	YES	4C22	4C29	4C30	NOTE 2
1	01111	YES	*	1	10000	NO	* 0	10001	NO	4C37	4C3E	4C45	NOTE 2
1	10010	NO	*	1	10011	NO	* 0	10100	NO	4C4C	4C53	4C5A	NOTE 2
0	10101	NO	*	X	10110	YES	* X	10111	YES	4C61	4C68	4C6F	NOTE 2
X	11000	YES	*	X	11001	YES	* X	11010	YES	4C76	4C7D	4C84	NOTE 2
X	11011	YES	*	X	11100	YES	* X	11101	YES	4C8B	4C92	4C99	NOTE 2
X	11110	YES	*	X	11111	YES	*			4CA0	4CA7		NOTE 2

NOTE 1: EACH BIT HAS ITS OWN UNIQUE MEMORY BYTE ASSIGN. 4BBE=MSB 4BC7=LSB 1-31H,0=30H,X=32H

NOTE 2: NO=30H, YES=31H

```
                                        00:09:50
                                        HEX 1553
   DISPLAY CAPTURE                       PAGE 02

TRIG MODE=START  ..........................................4AC0 NOTE 1
TRIG LINK-T1 OR  T2 OR  T3 OR  T4  ..................4AC1 4AC2 4AC3 NOTE 2

TRIG ENA BUS  ERROR   VALUE          ENA  BUS  =/-  ERR  =/-  VAL
T1   YES BTH =       =CXX X XX XX --4AC4 4AC5 4AC6 4AC7 4AC8 4AC9
T2   NO  BTH =       =CXX X XX XX --4AE3 4AE4 4AE5 4AE6 4AE7 4AE8
T3   NO  BTH =       =CXX X XX XX --4B02 4B03 4B04 4B05 4B06 4B07
T4   NO  BTH =       =CXX X XX XX --4B21 4B22 4B23 4B24 4B25 4B26

CAPT MODE-ALL  ............................................4B40 NOTE 3

MESG ENA BUS  ERROR   VALUE          ENA  BUS  =/-  ERR  =/-  VAL
M1   NO  BTH =       =CXX X XX XX   4B41 4B42 4B43 4B44 4B45 4B46
M2   NO  BTH =       =CXX X XX XX   4B60 4B61 4B62 4B63 4B64 4B65
M3   NO  BTH =       =CXX X XX XX   4B7F 4B80 4B81 4B82 4B83 4B84
M4   NO  BTH =       =CXX X XX XX   4B9E 4B9F 4BA0 4BA1 4BA2 4BA3
```

NOTE 1: START=30H,CENT=31H,END=32H

NOTE 2: OR=30H,ARM=31H,RST=32H

NOTE 3: ALL=30H,ONLY=31H

ENABLE: NO=30H,YES=31H

BUS: PRI=30H,SEC=31H,BTH=32H

=/- : "="=30H,"-"=31H

ERROR:				
	NO ERROR=30H	NORESP=31H	RMODE=32H	LOBIT=33H
	HIBIT=34H	PARITY=35H	TWOBUS=36H	INVSYNK=37H
	GAP=38H	WRDCNT=39H	ANYERR=41H	BITERR=42H
	MEBIT=43H	STABIT=44H	EARLY=45H	

=/- ELEVEN BYTE ASCII FIELD. ADDRESS SHOWN IS MSB

I/O
SETUP

```
                                        00:08:17
                                        HEX 1553
  DISPLAY I/O SETUP                      PAGE 05

  SERIAL INTERFACE:

BAUD RATE-------------------------0  ---------------------4AB9H   NOTE 1
DATA BITS-------------------------8  ---------------------4ABAH   NOTE 2
STOP BITS (A=1.B=1.5)-------------A  ---------------------4ABBH   NOTE 3
PARITY ENABLE---------------------NO  --------------------4ABCH   NOTE 4
PARITY----------------------------ODD  -------------------4ABDH   NOTE 5
BAUD SEL:  0=9600 1=110 2=150 3=300 4=600
           5=1200 6=2400 7=4800 8=19200

GPIB INTERFACE:

TERM ADDRESS----------------------30  --------------------4ABEH   NOTE 6
```

NOTE 1: 30H=9600 31H=110
32H=150 33H=300
34H=600 35H=1200
36H=2400 37H=4800
38H=19200

NOTE 2: 30H=8 31H=7
32H=6 33H=5

NOTE 3: 30H=B 31H=A

NOTE 4: 30H=NO 31H=YES

NOTE 5: 30H=ODD 31H=EVEN

NOTE 6: SIXTEEN BIT ASCII HEX FIELD
4ABEH=MSB
4ABFH=LSB

FRAME FORMAT

```
                                                00:06:44
                                                HEX 1553
  DISPLAY FRAME FORMAT                           PAGE 10

MINOR FRAME TIME ----------------0010 MSEC    ----------50A8H  NOTE 1
000   //  //  //  //  //  //  //  //  //  //  ----------50B0H  NOTE 2
010   //  //  //  //  //  //  //  //  //  //
020   //  //  //  //  //  //  //  //  //  //
030   //  //  //  //  //  //  //  //  //  //
040   //  //  //  //  //  //  //  //  //  //
050   //  //  //  //  //  //  //  //  //  //
060   //  //  //  //  //  //  //  //  //  //
070   //  //  //  //  //  //  //  //  //  //
080   //  //  //  //  //  //  //  //  //  //
090   //  //  //  //  //  //  //  //  //  //
100   //  //  //  //  //  //  //  //  //  //
110   //  //  //  //  //  //  //  //  //  //
120   //  //  //  //  //  //  //  **
```

NOTE 1: FOUR BYTE ASCII DECIMAL FIELD. 50A8H=MSB 50AB=LSB

NOTE 2: A 128 BYTE BUFFER BEGINNING AT 50B0H. [//]=00. [**]=0FEH. [--]=0FFH OR BINARY VALUE OF DESIRED BC.

03:56:26
HEX 1553
DISPLAY PRIORITY PAGE 13

DEFAULT	PRIORITY		SETUP		
NORESP	01	HIGHEST	NORESP	40B4	NOTE 1
BITERR	02		BITERR	40B5	NOTE 1
LOBIT	03		LOBIT	40B6	NOTE 1
PARITY	04		PARITY	40B7	NOTE 1
TWOBUS	04		TWOBUS	40B8	NOTE 1
INVSYC	06		INVSYC	40B9	NOTE 1
MEBIT	07		MEBIT	40BA	NOTE 1
STABIT	08		STABIT	40BB	NOTE 1
EARLY	09		EARLY	40BC	NOTE 1
WRDCNT	10		WRDCNT	40BD	NOTE 1
GAP	11		GAP	40BE	NOTE 1
RMODE	12		RMODE	40BF	NOTE 1
HIBIT	13	LOWEST	HIBIT	40C0	NOTE 1

NOTE 1: ALL ERRORS MUST BE ENTERED ONCE AND ONLY ONCE. NO DUPLICATIONS OR OMISSIONS ARE ALLOWED

41H=NORESP
42H=BITERR
33H=LOBIT
35H=PARITY
36H=TWOBUS
37H=INVSYC
43H=MEBIT
44H=STABIT
45H=EARLY
39H=WRDCNT
38H=GAP
32H=RMODE
34H=HIBIT

```
                                          00:01:12
                                          HEX 1553
   DISPLAY BC 01                             BC 01

CMD WORD-------------C01 R 01 01  ---------------------------4CA9 NOTE 1
GAP TIME-------------------------0010 USEC  ----------------4CAB NOTE 2
BUS SELECT-----------------------------PRI  ----------------4CB1 NOTE 3
SBA RESPONSE---------------------------YES  ----------------4CA8 NOTE 4
ERROR:  -------------------------------------------------------4CB1 NOTE 5
FUNCTION:  ----------------------------------------------------4CB4 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  --------------------------------------------------------4CAF NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F.
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4CA9H=MSB
4CAAH=LSB

NOTE 2: ON ALL BC PAGES, THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.
4CABH=LSB
4CACH=2ND LSB
4CADH=3RD LSB
4CAEH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4CB4H = 2ND MSB
4CB5H = 2ND GROUP 8 BITS
4CB6H = 3RD GROUP 8 BITS
4CB7H = 4TH GROUP 8 BITS

NOTE 7: SIXTEEN-BIT BINARY FIELD. 667 MAX.
4CAFH=LSB
4CB0H=MSB

NOTE 8: DATA BUFFER IS LOCATED AT C302 TO C87

BC 02

```
                                        00:01:12
                                        HEX 1553
   DISPLAY BC 02                           BC 02

CMD WORD-------------C00 R 00 00 ---------------------------4CB9 NOTE 1
GAP TIME-------------------------0001 MSEC ----------------4CBB NOTE 2
BUS SELECT-----------------------------PRI ----------------4CC1 NOTE 3
SBA RESPONSE---------------------------NO ----------------4CB8 NOTE 4
ERROR: ---------------------------------------------------4CC1 NOTE 5
FUNCTION: ------------------------------------------------4CC4 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA: ----------------------------------------------------4CBF NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4CB9H=MSB
4CBAH-LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4CBBH=LSB
4CBCH=2ND LSB
4CBDH=3RD LSB
4CBEH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4CC4H = 2ND MSB
4CC5H = 2ND GROUP 8 BITS
4CC6H = 3RD GROUP 8 BITS
4CC7H = 4TH GROUP 8 BITS

NOTE 7: SIXTEEN BIT BINARY FIELD. 667 MAX.
4CBFH=LSB
4CC0H=MSB

BC 03

```
                                        00:01:12
                                        HEX 1553
   DISPLAY BC 03                           BC 03

CMD WORD------------C00 R 00 00  ----------------------------4CC9 NOTE 1
GAP TIME----------------------------0001 MSEC  ---------------4CCB NOTE 2
BUS SELECT--------------------------------PRI  ---------------4CD1 NOTE 3
SBA RESPONSE------------------------------NO   ---------------4CC8 NOTE 4
ERROR:  ----------------------------------------------------4CD1 NOTE 5
FUNCTION:  -------------------------------------------------4CD4 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------------------------4CCF NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4CC9H=MSB
4CCAH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4CCBH=LSB
4CCCH=2ND LSB
4CCDH=3RD LSB
4CCEH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4CD4H=2ND MSB
4CD5H=MSB
4CD6H=LSB
4CD7H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD 667 MAX.
4CCFH=LSB
4CD0H=MSB

BC 04

```
                                        00:01:12
                                        HEX 1553
   DISPLAY BC 04                           BC 04

CMD WORD-------------C00 R 00 00 ----------------------------4CD9 NOTE 1
GAP TIME-------------------------0001 MSEC ------------------4CDB NOTE 2
BUS SELECT-----------------------------PRI ------------------4CE1 NOTE 3
SBA RESPONSE---------------------------NO -------------------4CD8 NOTE 4
ERROR: ---------------------------------------------------- 4CE1 NOTE 5
FUNCTION: ------------------------------------------------- 4CE4 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA: ----------------------------------------------------- 4CCF NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4CD9H=MSB
4CDAH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4CDBH=LSB
4CDCH=2ND LSB
4CDDH=3RD LSB
4CDEH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4CE4H=2ND MSB
4CE5H=MSB
4CE6H=LSB
4CE7H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD. 667 MAX
4CDFH=LSB
4CE0H=MSB

BC 05

```
                                        00:01:12
                                        HEX 1553
   DISPLAY BC 05                           BC 05

CMD WORD-------------C00 R 00 00  ----------------------------4CE9 NOTE 1
GAP TIME-------------------------0001 MSEC  ------------------4CEB NOTE 2
BUS SELECT-----------------------------PRI  ------------------4CF1 NOTE 3
SBA RESPONSE---------------------------NO   ------------------4CE8 NOTE 4
ERROR:  --------------------------------------------------------4CF1 NOTE 5
FUNCTION:  -----------------------------------------------------4CF4 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ---------------------------------------------------------4CEF NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4CE9H=MSB
4CEAH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.
4CEBH=LSB
4CECH=2ND LSB
4CEDH=3RD LSB
4CEEH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4. FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3)

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4CF4H=2ND MSB
4CF5H=MSB
4CF6H=LSB
4CF7H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD. 667 MAX
4CFFH=LSB
4CF0H=MSB

```
                                          00:01:12
                                          HEX 1553
   DISPLAY BC 06                             BC 06

CMD WORD-------------C00 R 00 00 ------------------------------4CF9 NOTE 1
GAP TIME-----------------------------0001 MSEC ----------------4CFB NOTE 2
BUS SELECT---------------------------------PRI ----------------4C01 NOTE 3
SBA RESPONSE-------------------------------NO -----------------4CF8 NOTE 4
ERROR: -------------------------------------------------------4D01 NOTE 5
FUNCTION: ----------------------------------------------------4D04 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA: --------------------------------------------------------4CFF NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4CF9H=MSB
4CFAH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.
4CFBH=LSB
4CFCH=2ND LSB
4CFDH=3RD LSB
4CFEH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4D04H=2ND MSB
4D05H=MSB
4D06H=LSB
4D07H=2ND LSB

NOTE 7 SIXTEEN BIT BINARY FIELD 667 MAX.
4CFFH=LSB
4D00H=MSB

BC 07

```
                                                  00:01:12
                                                  HEX 1553
   DISPLAY BC 07                                     BC 07

CMD WORD-------------C00 R 00 00  ----------------------------4D09 NOTE 1
GAP TIME-------------------------0001 MSEC  ----------------4D0B NOTE 2
BUS SELECT-----------------------------PRI  ----------------4D11 NOTE 3
SBA RESPONSE---------------------------NO   ----------------4D08 NOTE 4
ERROR:  --------------------------------------------------------4D11 NOTE 5
FUNCTION:  -----------------------------------------------------4D14 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ---------------------------------------------------------4D0F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4D09H=MSB
4D0AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4D0BH=LSB
4D0CH=2ND LSB
4D0DH=3RD LSB
4D0EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)

4D14H=2ND MSB
4D15H=MSB
4D16H=LSB

4D12H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD 667 MAX
4D0FH=LSB
4D11H=MSB

```
                                        00:01:12
                                        HEX 1553
   DISPLAY BC 08                           BC 08

CMD WORD-------------C00 R 00 00  ----------------------------4D19 NOTE 1
GAP TIME-------------------------0001 MSEC  ----------------4D1B NOTE 2
BUS SELECT-----------------------------PRI  ----------------4D21 NOTE 3
SBA RESPONSE---------------------------NO  ----------------4D18 NOTE 4
ERROR:  ------------------------------------------------------4D21 NOTE 5
FUNCTION:  ---------------------------------------------------4D24 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -------------------------------------------------------4D1F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4D19H=MSB
4D1AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4D1BH=LSB
4D1CH=2ND LSB
4D1DH=3RD LSB
4D1EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4D24H=2ND MSB
4D25H=MSB
4D26H=LSB
4D27H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD 667 MAX
4D1FH=LSB
4D20H=MSB

```
                                                    HEX 1553
   DISPLAY BC 09                                      BC 09

CMD WORD-------------C00 R 00 00  ----------------------------4D29 NOTE 1
GAP TIME-------------------------0001 MSEC  ------------------4D2B NOTE 2
BUS SELECT-----------------------------PRI  ------------------4D31 NOTE 3
SBA RESPONSE---------------------------NO  -------------------4D28 NOTE 4
ERROR:  --------------------------------------------------------4D31 NOTE 5
FUNCTION:  -----------------------------------------------------4D34 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ---------------------------------------------------------4D2F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4D29H=MSB
4D2AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4D2BH=LSB
4D2CH=2ND LSB
4D2DH=3RD LSB
4D2EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4D34H=2ND MSB
4D35H=MSB
4D36H=LSB
4D37H=2ND LSB

NOTE 7: SIX~~TEEN~~ BIT BINARY FIELD 667 MAX
4D2FH=LSB
4D30H=MSB

```
                                        00:01:12
                                        HEX 1553
   DISPLAY BC 10                           BC 10

CMD WORD-------------C00 R 00 00  ----------------------------4D39 NOTE 1
GAP TIME-------------------------0001 MSEC  ----------------4D3B NOTE 2
BUS SELECT-------------------------------PRI  ----------------4D41 NOTE 3
SBA RESPONSE-----------------------------NO   ----------------4D38 NOTE 4
ERROR:  ----------------------------------------------------4D41 NOTE 5
FUNCTION:  -------------------------------------------------4D44 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------------------------4D3F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4D39H=MSB
4D3AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4D3BH=LSB
4D3CH=2ND LSB
4D3DH=3RD LSB
4D3EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4D44H=2ND MSB
4D45H=MSB
4D46H=LSB
4D47H=2ND LSB

NOTE 7 SIXTEEN BIT BINARY FIELD. 667 MAX 4D3FH=LSB 4D40H=MSB

```
                                          00:01:12
                                          HEX 1553
   DISPLAY BC 11                             BC 11

CMD WORD-------------C00 R 00 00  ----------------------------4D49 NOTE 1
GAP TIME-------------------------0001 MSEC  ----------------4D4B NOTE 2
BUS SELECT-----------------------------PRI  ----------------4D51 NOTE 3
SBA RESPONSE---------------------------NO  ----------------4D48 NOTE 4
ERROR:  --------------------------------------------------4D51 NOTE 5
FUNCTION:  -----------------------------------------------4D54 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ---------------------------------------------------4D4F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4D49H=MSB
4D4AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4D4BH=LSB
4D4CH=2ND LSB
4D4DH=3RD LSB
4D4EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4D54H=2ND MSB
4D55H=MSB
4D56H=LSB
4D57H=2ND LSB

NOTE 7 SIX~~TEEN~~ BIT BINARY FIELD. 667 MAX.
4D4FH=LSB
4D50H=MSB

BC 12

```
                                                00:01:12
                                                HEX 1553
   DISPLAY BC 12                                   BC 12

CMD WORD-------------C00 R 00 00  ------------------------------4D59 NOTE 1
GAP TIME-------------------------0001 MSEC  ------------------4D5B NOTE 2
BUS SELECT-----------------------------PRI  ------------------4D61 NOTE 3
SBA RESPONSE---------------------------NO  -------------------4D58 NOTE 4
ERROR:  --------------------------------------------------------4D61 NOTE 5
FUNCTION:  -----------------------------------------------------4D64 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ---------------------------------------------------------4D5F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4D59H=MSB
4D5AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4D5BH=LSB
4D5CH=2ND LSB
4D5DH=3RD LSB
4D5EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4D64H=2ND MSB
4D65H=MSB
4D66H=LSB
4D67H=2ND LSB

NOTE 7. SIXTEEN BIT BINARY FIELD. 667 MAX.
4D5FH=LSB
4D60H=MSB

BC 13

```
                                        00:01:12
                                        HEX 1553
   DISPLAY BC 13                          BC 13

CMD WORD-------------C00 R 00 00 ------------------------------4D69 NOTE 1
GAP TIME-------------------------0001 MSEC ------------------4D6B NOTE 2
BUS SELECT-----------------------------PRI ------------------4D71 NOTE 3
SBA RESPONSE---------------------------NO ------------------4D68 NOTE 4
ERROR: ---------------------------------------------------------4D71 NOTE 5
FUNCTION: ------------------------------------------------------4D74 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA: ----------------------------------------------------------4D6F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4D69H=MSB
4D6AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.
4D6BH=LSB
4D6CH=2ND LSB
4D6DH=3RD LSB
4D6EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4D74H=2ND MSB
4D75H=MSB
4D76H=LSB
4D77H=2ND LSB

NOTE 7. SIX~~TEEN~~ BIT BINARY FIELD. 667 MAX
4D6FH=LSB
4D70H=MSB

BC 14

```
                                        00:01:12
                                        HEX 1553
   DISPLAY BC 14                           BC 14

CMD WORD.............C00 R 00 00  ..............................4D79 NOTE 1
GAP TIME.............................0001 MSEC  ..................4D7B NOTE 2
BUS SELECT.....................................PRI  ..............4D81 NOTE 3
SBA RESPONSE...................................NO  ..............4D78 NOTE 4
ERROR:  ...........................................................4D81 NOTE 5
FUNCTION:  ........................................................4D84 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ............................................................4D7F NOTE 7
0000 ........0000     0001     0002     0003
0004 ........0004     0005     0006     0007
0008 ........0008     0009     000A     000B
0012 ........000C     000D     000E     000F
0016 ........0010     0011     0012     0013
0020 ........0014     0015     0016     0017
0024 ........0018     0019     001A     001B
0028 ........001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4D79H=MSB
4D7AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4D7BH=LSB
4D7CH=2ND LSB
4D7DH=3RD LSB
4D7EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3)

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4D84H=2ND MSB
4D85H=MSB
4D86H=LSB
4D87H=2ND LSB

NOTE 7 SIXTEEN BIT BINARY FIELD. 667 MAX.
4D7FH=LSB
4D80H=MSB

```
                                        00:01:12
                                        HEX 1553
   DISPLAY BC 15                           BC 15

CMD WORD.............C00 R 00 00  ..............................4D89 NOTE 1
GAP TIME.............................0001 MSEC  ..................4D8B NOTE 2
BUS SELECT.................................PRI  ..................4D91 NOTE 3
SBA RESPONSE...............................NO  ..................4D88 NOTE 4
ERROR:  ..................................................... ...4D91 NOTE 5
FUNCTION:  ..................................................... ..4D94 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ..................................................... ...4D8F NOTE 7
0000 ........0000     0001     0002     0003
0004 ........0004     0005     0006     0007
0008 ........0008     0009     000A     000B
0012 ........000C     000D     000E     000F
0016 ........0010     0011     0012     0013
0020 ........0014     0015     0016     0017
0024 ........0018     0019     001A     001B
0028 ........001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD. 4D89H=MSB 4D8AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4D8BH=LSB
4D8CH=2ND LSB
4D8DH=3RD LSB
4D8EH=MSB

NOTE 3. FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4D94H=2ND MSB
4D95H=MSB
4D96H=LSB
4D97H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD. 667 MAX.
4D8FH=LSB

4D~~90H~~=MSB

```
                                          00:01:12
                                          HEX 1553
   DISPLAY BC 16                             BC 16

CMD WORD-------------C00 R 00 00  ----------------------------4D99 NOTE 1
GAP TIME-------------------------0001 MSEC  ----------------4D9B NOTE 2
BUS SELECT-------------------------------PRI  ----------------4DA1 NOTE 3
SBA RESPONSE-----------------------------NO  ----------------4D98 NOTE 4
ERROR:  ---------------------------------------------------------4DA1 NOTE 5
FUNCTION:  ------------------------------------------------------4DA4 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------------4D9F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4D99H=MSB
4D9AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4D9BH=LSB
4D9CH=2ND LSB
4D9DH=3RD LSB
4D9EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4DA4H=2ND MSB
4DA5H=MSB
4DA6H=LSB

4DA7H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD. 667 MAX
4D9FH=LSB
4DA0H=MSB

```
                                          00:01:12
                                          HEX 1553
   DISPLAY BC 17                             BC 17

CMD WORD-------------C00 R 00 00  ----------------------------4DA9 NOTE 1
GAP TIME-------------------------0001 MSEC  ----------------4DAB NOTE 2
BUS SELECT-------------------------------PRI  ----------------4DB1 NOTE 3
SBA RESPONSE-----------------------------NO  ----------------4DA8 NOTE 4
ERROR:  ----------------------------------------------------4DB1 NOTE 5
FUNCTION:  ------------------------------------------------4DB4 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------4DAF NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4DA9H=MSB
4DAAH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4DABH=LSB
4DACH=2ND LSB
4DADH=3RD LSB
4DAEH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4DB4H=2ND MSB
4DB5H=MSB
4DB6H=LSB

4DB7H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD 667 MAX
4DAFH=LSB
4DB0H=MSB

BC 18

```
                                                00:01:12
                                                HEX 1553
   DISPLAY BC 18                                   BC 18

CMD WORD-------------C00 R 00 00  ------------------------------4DB9 NOTE 1
GAP TIME-------------------------0001 MSEC  ------------------4DBB NOTE 2
BUS SELECT-----------------------------PRI  ------------------4DC1 NOTE 3
SBA RESPONSE---------------------------NO   ------------------4DB8 NOTE 4
ERROR:  --------------------------------------------------------4DC1 NOTE 5
FUNCTION:  -----------------------------------------------------4DC4 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ---------------------------------------------------------4DBF NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4DB9H=MSB
4DBAH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC

4DBBH=LSB
4DBCH=2ND LSB
4DBDH=3RD LSB
4DBEH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4DC4H=2ND MSB
4DC5H=MSB
4DC6H=LSB
4DC7H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD. 667 MAX.
4DBFH=LSB
4DC0H=MSB

BC 19

```
                                        00:01:12
                                        HEX 1553
   DISPLAY BC 19                           BC 19

CMD WORD-----------C00 R 00 00 ----------------------------4DC9 NOTE 1
GAP TIME-----------------------------0001 MSEC ------------4DCB NOTE 2
BUS SELECT---------------------------PRI ------------------4DD1 NOTE 3
SBA RESPONSE-------------------------NO -------------------4DC8 NOTE 4
ERROR: -------------------------------------------------- 4DD1 NOTE 5
FUNCTION: ----------------------------------------------- 4DD4 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA: --------------------------------------------------- 4DCF NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4DC9H=MSB
4DCAH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4DCBH=LSB
4DCCH=2ND LSB
4DCDH=3RD LSB
4DCEH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4DD4H=2ND MSB
4DD5H=MSB
4DD6H=LSB
4DD7H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD. 667 MAX.
4DCFH=LSB
4DD0H=MSB

```
                                                00:01:12
                                                HEX 1553
   DISPLAY BC 20                                   BC 20

CMD WORD-------------C00 R 00 00  ----------------------------4DD9 NOTE 1
GAP TIME-------------------------0001 MSEC  ------------------4DDB NOTE 2
BUS SELECT-----------------------------PRI  ------------------4DE1 NOTE 3
SBA RESPONSE---------------------------NO   ------------------4DD8 NOTE 4
ERROR:  --------------------------------------------------------4DE1 NOTE 5
FUNCTION:  -----------------------------------------------------4DE4 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ---------------------------------------------------------4DDF NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4DD9H=MSB
4DDAH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4DDBH=LSB
4DDCH=2ND LSB
4DDDH=3RD LSB
4DDEH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION. 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4DE4H=2ND MSB
4DE5H=MSB
4DE6H=LSB
4DE7H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD. 667 MAX.
4DDFH=LSB
4DE0H=MSB

```
                                        00:01:12
                                        HEX 1553
   DISPLAY BC 21                           BC 21

CMD WORD.............C00 R 00 00  ..............................4DE9 NOTE 1
GAP TIME.............................0001 MSEC  ..................4DEB NOTE 2
BUS SELECT.....................................PRI  ..............4DF1 NOTE 3
SBA RESPONSE...................................NO  ..............4DF8 NOTE 4
ERROR:  .......................................................4DF1 NOTE 5
FUNCTION:  ....................................................4DF4 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ........................................................4DEF NOTE 7
0000 .........0000     0001     0002     0003
0004 .........0004     0005     0006     0007
0008 .........0008     0009     000A     000B
0012 .........000C     000D     000E     000F
0016 .........0010     0011     0012     0013
0020 .........0014     0015     0016     0017
0024 .........0018     0019     001A     001B
0028 .........001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4DE9H=MSB
4DEAH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4DEBH=LSB
4DECH=2ND LSB
4DEDH=3RD LSB
4DEEH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4DF4H=2ND MSB
4DF5H=MSB
4DF6H=LSB
4DF7H=2ND LSB

NOTE 7: SIX~~TEEN~~ BIT BINARY FIELD. 667 MAX
4DEFH=LSB
4DF0H=MSB

BC 22

```
                                        00:01:12
                                        HEX 1553
   DISPLAY BC 22                           BC 22

CMD WORD-------------C00 R 00 00  ----------------------------4DF9 NOTE 1
GAP TIME-------------------------0001 MSEC  ----------------4DFB NOTE 2
BUS SELECT-----------------------------PRI  ----------------4E01 NOTE 3
SBA RESPONSE---------------------------NO   ----------------4DF8 NOTE 4
ERROR:  --------------------------------------------------------4E01 NOTE 5
FUNCTION:  -----------------------------------------------------4E04 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ---------------------------------------------------------4DFF NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4DF9H=MSB
4DFAH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4DFBH=LSB
4DFCH=2ND LSB
4DFDH=3RD LSB
4DFEH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4E04H=2ND MSB
4E05H=MSB
4E06H=LSB
4E07H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD. 667 MAX.
4DFFH=LSB
4E00H=MSB

BC 23

```
                                                00:01:12
                                                HEX 1553
   DISPLAY BC 23                                   BC 23

CMD WORD-------------C00 R 00 00  ----------------------------4E09 NOTE 1
GAP TIME-------------------------0001 MSEC  ------------------4E0B NOTE 2
BUS SELECT-----------------------------PRI  ------------------4E11 NOTE 3
SBA RESPONSE---------------------------NO   ------------------4E08 NOTE 4
ERROR:  --------------------------------------------------------4E11 NOTE 5
FUNCTION:  -----------------------------------------------------4E14 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ---------------------------------------------------------4E0F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4E09H=MSB
4E0AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4E0BH=LSB
4E0CH=2ND LSB
4E0DH=3RD LSB
4E0EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4E14H=2ND MSB
4E15H=MSB
4E16H=LSB
4E17H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD 667 MAX.
4E0FH=LSB
4E10H=MSB

```
                                                00:01:12
                                                HEX 1553
   DISPLAY BC 24                                   BC 24

CMD WORD-------------C00 R 00 00  ----------------------------4E19 NOTE 1
GAP TIME-----------------------------0001 MSEC  ---------------4E1B NOTE 2
BUS SELECT-----------------------------PRI  ------------------4E21 NOTE 3
SBA RESPONSE---------------------------NO  -------------------4E18 NOTE 4
ERROR:  ------------------------------------------------------4E21 NOTE 5
FUNCTION:  ---------------------------------------------------4E24 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -------------------------------------------------------4E1F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4E19H=MSB
4E1AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4E1BH=LSB
4E1CH=2ND LSB
4E1DH=3RD LSB
4E1EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4E24H=2ND MSB
4E25H=MSB
4E26H=LSB
4E27H=2ND LSB

NOTE 7: SIX~~TEE~~N BIT BINARY FIELD. 667 MAX.
4E1FH=LSB
4E20H=MSB

BC 25

```
                                                00:01:12
                                                HEX 1553
   DISPLAY BC 25                                   BC 25

CMD WORD-------------C00 R 00 00  ------------------------------4E29 NOTE 1
GAP TIME-------------------------0001 MSEC  ------------------4E2B NOTE 2
BUS SELECT-----------------------------PRI  ------------------4E31 NOTE 3
SBA RESPONSE---------------------------NO   ------------------4E28 NOTE 4
ERROR:  --------------------------------------------------------4E31 NOTE 5
FUNCTION:  -----------------------------------------------------4E34 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ---------------------------------------------------------4E2F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4E29H=MSB
4E2AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4E2BH=LSB
4E2CH=2ND LSB
4E2DH=3RD LSB
4E2EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4E34H=2ND MSB
4E35H=MSB
4E36H=LSB
4E37H=2ND LSB

NOTE 7 SIXTEEN BIT BINARY FIELD 667 MAX.
4E2FH=LSB
4E30H=MSB

BC 26

```
                                          00:01:12
                                          HEX 1553
   DISPLAY BC 26                             BC 26

CMD WORD-------------C00 R 00 00 ------------------------------4E39 NOTE 1
GAP TIME-----------------------------0001 MSEC ----------------4E3B NOTE 2
BUS SELECT---------------------------------PRI ----------------4E41 NOTE 3
SBA RESPONSE-------------------------------NO ----------------4E38 NOTE 4
ERROR: -------------------------------------------------------4E41 NOTE 5
FUNCTION: ----------------------------------------------------4E44 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA: --------------------------------------------------------4E3F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4E39H=MSB
4E3AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4E3BH=LSB
4E3CH=2ND LSB
4E3DH=3RD LSB
4E3EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4E44H=2ND MSB
4E45H=MSB
4E46H=LSB
4E47H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD 667 MAX.
4E3FH=LSB
4E40H=MSB

BC 27

```
                                                00:01:12
                                                HEX 1553
   DISPLAY BC 27                                   BC 27

CMD WORD-------------C00 R 00 00  ------------------------------4E49 NOTE 1
GAP TIME-----------------------------0001 MSEC  ---------------4E4B NOTE 2
BUS SELECT---------------------------------PRI  ---------------4E51 NOTE 3
SBA RESPONSE-------------------------------NO   ---------------4E48 NOTE 4
ERROR:  ------------------------------------------------------4E51 NOTE 5
FUNCTION:  ---------------------------------------------------4E54 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -------------------------------------------------------4E4F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4E49H=MSB
4E4AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4E4BH=LSB
4E4CH=2ND LSB
4E4DH=3RD LSB
4E4EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4E54H=2ND MSB
4E55H=MSB
4E56H=LSB
4E57H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD. 667 MAX
4E4FH=LSB
4E50H=MSB

BC 28

```
                                                00:01:12
                                                HEX 1553
   DISPLAY BC 28                                   BC 28

CMD WORD-------------C00 R 00 00  ------------------------------4E59 NOTE 1
GAP TIME-------------------------0001 MSEC  ------------------4E5B NOTE 2
BUS SELECT-----------------------------PRI  ------------------4E61 NOTE 3
SBA RESPONSE---------------------------NO   ------------------4E58 NOTE 4
ERROR:  --------------------------------------------------------4E61 NOTE 5
FUNCTION:  -----------------------------------------------------4E64 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ---------------------------------------------------------4E5F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4E59H=MSB
4E5AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4E5BH=LSB
4E5CH=2ND LSB
4E5DH=3RD LSB
4E5EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4E64H=2ND MSB
4E65H=MSB
4E66H=LSB
4E67H=2ND LSB

NOTE 7 SIXTEEN BIT BINARY FIELD. 667 MAX.
4E5FH=LSB
4E60H=MSB

```
                                                  00:01:12
                                                  HEX 1553
   DISPLAY BC 29                                     BC 29

CMD WORD-------------C00 R 00 00  ----------------------------4E69 NOTE 1
GAP TIME-------------------------0001 MSEC  ------------------4E6B NOTE 2
BUS SELECT-----------------------------PRI  ------------------4E71 NOTE 3
SBA RESPONSE---------------------------NO   ------------------4E68 NOTE 4
ERROR:  --------------------------------------------------------4E71 NOTE 5
FUNCTION:  -----------------------------------------------------4E74 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ---------------------------------------------------------4E6F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4E69H=MSB
4E6AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4E6BH=LSB
4E6CH=2ND LSB
4E6DH=3RD LSB
4E6EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4E74H=2ND MSB
4E75H=MSB
4E76H=LSB
4E77H=2ND LSB

NOTE 7. SIXTEEN BIT BINARY FIELD. 667 MAX
4E6FH=LSB
4E70H=MSB

BC 30

```
                                                00:01:12
                                                HEX 1553
   DISPLAY BC 30                                   BC 30

CMD WORD-------------C00 R 00 00  ------------------------------4E79 NOTE 1
GAP TIME-------------------------0001 MSEC  ------------------4E7B NOTE 2
BUS SELECT-----------------------------PRI  ------------------4E81 NOTE 3
SBA RESPONSE---------------------------NO  ------------------4E78 NOTE 4
ERROR:  --------------------------------------------------------4E81 NOTE 5
FUNCTION:  -----------------------------------------------------4E84 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ---------------------------------------------------------4E7F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD
4E79H=MSB
4E7AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4E7BH=LSB
4E7CH=2ND LSB
4E7DH=3RD LSB
4E7EH=MSB

NOTE 3: FORMAT IS BBBEEEEE BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3)

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4E84H=2ND MSB
4E85H=MSB
4E86H=LSB
4E87H=2ND LSB

NOTE 7 SIXTEEN BIT BINARY FIELD 667 MAX 4E7FH=LSB 4E80H=MSB

BC 31

```
                                          00.01:12
                                          HEX 1553
   DISPLAY BC 31                             BC 31

CMD WORD-------------C00 R 00 00  ------------------------------4E89 NOTE 1
GAP TIME-----------------------------0001 MSEC  ---------------4E8B NOTE 2
BUS SELECT---------------------------------PRI  ---------------4E91 NOTE 3
SBA RESPONSE-------------------------------NO  ---------------4E88 NOTE 4
ERROR:  ----------------------------------------------------------4E91 NOTE 5
FUNCTION:  -------------------------------------------------------4E94 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------------------------------4E8F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4E89H=MSB
4E8AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4, BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4E8BH=LSB
4E8CH=2ND LSB
4E8DH=3RD LSB
4E8EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION: 000=PRI 010=SEC

NOTE 4. FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4E94H=2ND MSB
4E95H=MSB
4E96H=LSB
4E97H=2ND LSB

NOTE 7: SIXTEEN BIT BINARY FIELD. 667 MAX.
4E8FH=LSB
4E90H=MSB

BC 32

```
                                                00:01:12
                                                HEX 1553
   DISPLAY BC 32                                   BC 32

CMD WORD-------------C00 R 00 00  ----------------------------4E99 NOTE 1
GAP TIME-----------------------------0001 MSEC  ---------------4E9B NOTE 2
BUS SELECT-----------------------------PRI   ----------------4EA1 NOTE 3
SBA RESPONSE---------------------------NO    ----------------4E98 NOTE 4
ERROR:  ------------------------------------------------------4EA1 NOTE 5
FUNCTION:  ---------------------------------------------------4EA4 NOTE 6
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -------------------------------------------------------4E9F NOTE 7
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: SIXTEEN BIT BINARY FIELD.
4E99H=MSB
4E9AH=LSB

NOTE 2: THIRTY-TWO BIT BINARY FIELD. THE FORMAT FOR DISPLAY MAY BE SELECTED ACCORDING TO NOTE 4. BUT THE ACTUAL VALUE IN THESE 4 BYTES ARE ALWAYS IN USEC.

4E9BH=LSB
4E9CH=2ND LSB
4E9DH=3RD LSB
4E9EH=MSB

NOTE 3: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 5 FOR EEEEE. BBB DESCRIPTION 000=PRI 010=SEC

NOTE 4: FORMAT IS UUUURRRR
UUUU
0100 = 4 GAPTIME IS IN USEC.
0000 = 0 GAPTIME IS IN MSEC.

RRRR
0000 = 0 No - SBA response disabled
0001 = 1 Yes - SBA response enabled

NOTE 5: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 6: 32 BIT BINARY FIELD - BIT FOR BIT COPY OF FUNCTION LINE (GROUPS DEFINED FROM LEFT TO RIGHT)
4EA4H=2ND MSB
4EA5H=MSB
4EA6H=LSB
4EA7H=2ND LSB

NOTE 7. SIXTEEN BIT BINARY FIELD 667 MAX
4E9FH=LSB
4EA0H=MSB

RT 01

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 01                                   RT 01

ENAB-------------------------------------NO  ----------------4EA8 NOTE 1
RESP BUS---------------------------------NOR ----------------4EB1 NOTE 2
RESP CMD-----------C01 R XX XX  ----------4EA9 NOTE 3 & 4EAB NOTE 4
STATUS WORD--------S01 0 00 00  --------------------------4EAD NOTE 5
ERROR:  ---------------------------------------------------4EB1 NOTE 6
FUNCTION:  ------------------------------------------------4EA4 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------4E9F NOTE 8
0000 --------0000    0001    0002    0003
0004 --------0004    0005    0006    0007
0008 --------0008    0009    000A    000B
0012 --------000C    000D    000E    000F
0016 --------0010    0011    0012    0013
0020 --------0014    0015    0016    0017
0024 --------0018    0019    001A    001B
0028 --------001C    001D    001E    001F
```

NOTE 1: 00=NO, 80H=YES

NOTE 2: ON ALL RT AND BC PAGES, FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE BBB DESCRIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD BINARY
4EA9H=MSB
4EAAH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'
4EABH=MSB
4EACH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4EADH=MSB
4EAEH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4EB4H=2ND MSB 4EB6H=LSB
4EB5H=MSB 4EB7H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4EAFH=LSB
4EB0H=MSB

RT 02

```
                                        00:13:29
                                        HEX 1553
   DISPLAY RT 02                           RT 02

ENAB-------------------------------------NO  ----------------4EB8 NOTE 1
RESP BUS---------------------------------NOR ----------------4EC1 NOTE 2
RESP CMD----------C00 R XX XX  ----------4EB9 NOTE 3 & 4EBB NOTE 4
STATUS WORD-------S00 0 00 00  --------------------------4EBD NOTE 5
ERROR:  ------------------------------------------------4EC1 NOTE 6
FUNCTION:  ---------------------------------------------4EC4 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -------------------------------------------------4E9F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
4EB9H=MSB
4EBAH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4EBBH=MSB
4EBCH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4EBDH=MSB
4EBEH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4EC4H=2ND MSB 4EC6H=LSB
4EC5H=MSB 4EC7H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4EBFH=LSB
4EC0H=MSB

```
                                                  00:13:29
                                                  HEX 1553
   DISPLAY RT 03                                     RT 03

ENAB-------------------------------------NO  ----------------4EC8 NOTE 1
RESP BUS---------------------------------NOR  ---------------4ED1 NOTE 2
RESP CMD-----------C00 R XX XX  ----------4EC9 NOTE 3 & 4ECB NOTE 4
STATUS WORD--------S00 0 00 00  --------------------------4ECD NOTE 5
ERROR:  ---------------------------------------------------4ED1 NOTE 6
FUNCTION:  ------------------------------------------------4ED4 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------4ECF NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY
4EC9H=MSB
4ECAH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4ECBH=MSB
4ECCH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4ECDH=MSB
4ECEH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4ED4H=2ND MSB 4ED6H=LSB
4ED5H=MSB 4ED7H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4ECFH=LSB
4ED0H=MSB

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 04                                   RT 04

ENAB-------------------------------------NO  ----------------4ED8 NOTE 1
RESP BUS---------------------------------NOR  ---------------4EE1 NOTE 2
RESP CMD-----------C00 R XX XX  ------------4ED9 NOTE 3 & 4EDB NOTE 4
STATUS WORD--------500 0 00 00  ----------------------------4EDD NOTE 5
ERROR:  ----------------------------------------------------4EE1 NOTE 6
FUNCTION:  -------------------------------------------------4EE4 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------------------------4EDF NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
4ED9H=MSB
4EDAH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4EDBH=MSB
4EDCH=LSB
NOTE 5: SIXTEEN BIT BINARY FIELD
4EDDH=MSB
4EDEH=LSB

NOTE 6. REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4EE4H=2ND MSB 4EE6H=LSB
4EE5H=MSB 4EE7H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4EDFH=LSB
4EE0H=MSB

```
                                        00:13:29
                                        HEX 1553
   DISPLAY RT 05                           RT 05

ENAB.....................................NO  .....................4EE8 NOTE 1
RESP BUS.................................NOR  ....................4EF1 NOTE 2
RESP CMD.............C00 R XX XX  ................4EE9 NOTE 3 & 4EEB NOTE 4
STATUS WORD..........S00 0 00 00  ...............................4EED NOTE 5
ERROR:  .................................................................4EF1 NOTE 6
FUNCTION:  ..............................................................4EF4 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ..................................................................4EEF NOTE 8
0000 .........0000     0001     0002     0003
0004 .........0004     0005     0006     0007
0008 .........0008     0009     000A     000B
0012 .........000C     000D     000E     000F
0016 .........0010     0011     0012     0013
0020 .........0014     0015     0016     0017
0024 .........0018     0019     001A     001B
0028 .........001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
4EE9H=MSB
4EEAH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL USE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD BE A 'DONT CARE'
4EEBH=MSB
4EECH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4EEDH=MSB
4EEEH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4EF4H=2ND MSB 4EF6H=LSB
4EF5H=MSB 4EF7H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4EEFH=LSB
4EF0H=MSB

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 06                                   RT 06

ENAB-------------------------------------NO  ----------------4EF8 NOTE 1
RESP BUS---------------------------------NOR  ---------------4F01 NOTE 2
RESP CMD-------------C00 R XX XX  ------------4EF9 NOTE 3 & 4EFB NOTE 4
STATUS WORD----------S00 0 00 00  ----------------------------4EFD NOTE 5
ERROR:  ------------------------------------------------------4F01 NOTE 6
FUNCTION:  ---------------------------------------------------4F04 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -------------------------------------------------------4EFF NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY
4EF9H=MSB
4EFAH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4EFBH=MSB
4EFCH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4EFDH=MSB
4EFEH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4F04H=2ND MSB 4F06H=LSB
4F05H=MSB 4F07H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4EFFH=LSB
4F00H=MSB

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 07                                   RT 07

ENAB-------------------------------------NO  ----------------4F08 NOTE 1
RESP BUS---------------------------------NOR  ---------------4F11 NOTE 2
RESP CMD-----------C00 R XX XX  ------------4F09 NOTE 3 & 4F0B NOTE 4
STATUS WORD--------500 0 00 00  ----------------------------4F0D NOTE 5
ERROR:  ----------------------------------------------------4F11 NOTE 6
FUNCTION:  -------------------------------------------------4F14 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------------------------4F0F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3. SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY
4F09H=MSB
4F0AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4F0BH=MSB
4F0CH=LSB

NOTE 5. SIXTEEN BIT BINARY FIELD.
4F0DH=MSB
4F0EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4F14H=2ND MSB 4F16H=LSB
4F15H=MSB 4F17H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4F0FH=LSB
4F10H=MSB

```
                                        00:13:29
                                        HEX 1553
   DISPLAY RT 08                           RT 08

ENAB.....................................NO  ..................4F18 NOTE 1
RESP BUS.................................NOR  .................4F21 NOTE 2
RESP CMD.............C00 R XX XX  ..............4F19 NOTE 3 & 4F1B NOTE 4
STATUS WORD..........S00 0 00 00  ..............................4F1D NOTE 5
ERROR:  ..................................................... 4F21 NOTE 6
FUNCTION:  ................................................... 4F24 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ....................................................... 4F1F NOTE 8
0000 ........0000     0001     0002     0003
0004 ........0004     0005     0006     0007
0008 ........0008     0009     000A     000B
0012 ........000C     000D     000E     000F
0016 ........0010     0011     0012     0013
0020 ........0014     0015     0016     0017
0024 ........0018     0019     001A     001B
0028 ........001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
4F19H=MSB
4F1AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4F1BH=MSB
4F1CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4F1DH=MSB
4F1EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4F24H=2ND MSB 4F26H=LSB
4F25H=MSB 4F27H=2ND LSB

NOTE 8: SIXTEEN
BIT BINARY FIELD. 667 MAX.
4F1FH=LSB
4F20H=MSB

RT 09

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 09                                   RT 09

ENAB-------------------------------------NO  ----------------4F28 NOTE 1
RESP BUS---------------------------------NOR ----------------4F31 NOTE 2
RESP CMD----------C00 R XX XX  ------------4F29 NOTE 3 & 4F2B NOTE 4
STATUS WORD-------S00 0 00 00  ----------------------------4F2D NOTE 5
ERROR:  ---------------------------------------------------4F31 NOTE 6
FUNCTION:  ------------------------------------------------4F34 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------4F2F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
4F29H=MSB
4F2AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4F2BH=MSB
4F2CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4F2DH=MSB
4F2EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4F34H=2ND MSB 4F36H=LSB
4F35H=MSB 4F37H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4F2FH=LSB
4F30H=MSB

```
                                        00:12:28
                                        HEX 1553
   DISPLAY RT 10                          RT 10

ENAB-------------------------------NO  ----------------4F38 NOTE 1
RESP BU----------------------------NOR  ---------------4F41 NOTE 2
RESP CMD----------C00 R XX XX  ------------4F29 NOTE 3 & 4F3B NOTE 4
STATUS WORD-------S00 0 00 00  ----------------------------4F3D NOTE 5
ERROR:  ---------------------------------------------------4F41 NOTE 6
FUNCTION:  ------------------------------------------------4F44 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------4F3F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD BINARY
4F39H=MSB
4F3AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4F3BH=MSB
4F3CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4F3DH=MSB
4F3EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4F44H=2ND MSB 4F46H=LSB
4F45H=MSB 4F47H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4F3FH=LSB
4F40H=MSB

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 11                                   RT 11

ENAB-------------------------------------NO  ----------------4F48 NOTE 1
RESP BUS---------------------------------NOR  ---------------4F51 NOTE 2
RESP CMD----------C00 R XX XX  ------------4F49 NOTE 3 & 4F4B NOTE 4
STATUS WORD-------S00 0 00 00  ----------------------------4F4D NOTE 5
ERROR:  ---------------------------------------------------4F51 NOTE 6
FUNCTION:  ------------------------------------------------4F54 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------4F4F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
4F49H=MSB
4F4AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4F4BH=MSB
4F4CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4F4DH=MSB
4F4EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4F54H=2ND MSB 4F56H=LSB
4F55H=MSB 4F57H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4F4FH=LSB
4F50H=MSB

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 12                                   RT 12

ENAB.....................................NO  ..................4F58 NOTE 1
RESP BUS.................................NOR  .................4F61 NOTE 2
RESP CMD.............C00 R XX XX  ..............4F59 NOTE 3 & 4F5B NOTE 4
STATUS WORD..........S00 0 00 00  ..............................4F5D NOTE 5
ERROR:  ..................................................4F61 NOTE 6
FUNCTION:  ...............................................4F64 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ...................................................4F5F NOTE 8
0000 ........0000     0001     0002     0003
0004 ........0004     0005     0006     0007
0008 ........0008     0009     000A     000B
0012 ........000C     000D     000E     000F
0016 ........0010     0011     0012     0013
0020 ........0014     0015     0016     0017
0024 ........0018     0019     001A     001B
0028 ........001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
4F59H=MSB
4F5AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4F5BH=MSB
4F5CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4F5DH=MSB
4F5EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4F64H=2ND MSB 4F66H=LSB
4F65H=MSB 4F67H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4F5FH=LSB
4F60H=MSB

RT 13

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 13                                   RT 13

ENAB-------------------------------------NO  ----------------4F68 NOTE 1
RESP BUS---------------------------------NOR ----------------4F71 NOTE 2
RESP CMD-----------C00 R XX XX  ------------4F69 NOTE 3 & 4F6B NOTE 4
STATUS WORD--------S00 0 00 00  ----------------------------4F6D NOTE 5
ERROR:  ----------------------------------------------------4F71 NOTE 6
FUNCTION:  -------------------------------------------------4F74 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------------------------4F6F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO, 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
4F69H=MSB
4F6AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4F6BH=MSB
4F6CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4F6DH=MSB
4F6EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4F74H=2ND MSB 4F76H=LSB
4F75H=MSB 4F77H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4F6FH=LSB
4F70H=MSB

RT 14

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 14                                   RT 14

ENAB.....................................NO  ..................4F78 NOTE 1
RESP BUS.................................NOR  .................4F81 NOTE 2
RESP CMD.............C00 R XX XX  .............4F79 NOTE 3 & 4F7B NOTE 4
STATUS WORD..........S00 0 00 00  ............................4F7D NOTE 5
ERROR: ..................................................................4F81 NOTE 6
FUNCTION: ...............................................................4F84 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA: ...................................................................4F7F NOTE 8
0000 ........0000     0001     0002     0003
0004 ........0004     0005     0006     0007
0008 ........0008     0009     000A     000B
0012 ........000C     000D     000E     000F
0016 ........0010     0011     0012     0013
0020 ........0014     0015     0016     0017
0024 ........0018     0019     001A     001B
0028 ........001C     001D     001E     001F
```

NOTE 1: 00=NO, 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY
4F79H=MSB
4F7AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4F7BH=MSB
4F7CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4F7DH=MSB
4F7EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4F84H=2ND MSB 4F86H=LSB
4F85H=MSB 4F87H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4F7FH=LSB
4F80H=MSB

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 15                                   RT 15

ENAB-------------------------------------NO  ----------------4F88 NOTE 1
RESP BUS---------------------------------NOR  ---------------4F91 NOTE 2
RESP CMD----------C00 R XX XX  ------------4F89 NOTE 3 & 4F8B NOTE 4
STATUS WORD-------S00 0 00 00  ----------------------------4F8D NOTE 5
ERROR:  ---------------------------------------------------4F91 NOTE 6
FUNCTION:  ------------------------------------------------4F94 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------4F8F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
4F89H=MSB
4F8AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4F8BH=MSB
4F8CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4F8DH=MSB
4F8EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4F94H=2ND MSB 4F96H=LSB
4F95H=MSB 4F97H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4F8FH=LSB
4F90H=MSB

RT 16

```
                                        00:13:29
                                        HEX 1553
   DISPLAY RT 16                           RT 16

ENAB-------------------------------------NO  ------------------4F98 NOTE 1
RESP BUS---------------------------------NOR  -----------------4FA1 NOTE 2
RESP CMD-----------C00 R XX XX  ----------4F99 NOTE 3 & 4F9B NOTE 4
STATUS WORD--------S00 0 00 00  ----------------------------4F9D NOTE 5
ERROR:  ----------------------------------------------------4FA1 NOTE 6
FUNCTION:  -------------------------------------------------4FA4 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------------------------4F9F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
4F99H=MSB
4F9AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4F9BH=MSB
4F9CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4F9DH=MSB
4F9EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4FA4H=2ND MSB 4FA6H=LSB
4FA5H=MSB 4FA7H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX. 4F9FH=LSB 4FA0H=MSB

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 17                                   RT 17

ENAB-------------------------------------NO  ----------------4FA8 NOTE 1
RESP BUS---------------------------------NOR ----------------4FB1 NOTE 2
RESP CMD-----------C00 R XX XX  ----------4FA9 NOTE 3 & 4FAB NOTE 4
STATUS WORD--------S00 0 00 00  ----------------------------4FAD NOTE 5
ERROR:  ----------------------------------------------------4FB1 NOTE 6
FUNCTION:  -------------------------------------------------4FB4 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------------------------4FAF NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION. 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
4FA9H=MSB
4FAAH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4FABH=MSB
4FACH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4FADH=MSB
4FAEH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4FB4H=2ND MSB 4FB6H=LSB
4FB5H=MSB 4FB7H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4FAFH=LSB
4FB0H=MSB

```
                                        00:13 29
                                        HEX 1553
   DISPLAY RT 18                           RT 18

ENAB-------------------------------------NO  ----------------4FB8 NOTE 1
RESP BUS---------------------------------NOR ----------------4FC1 NOTE 2
RESP CMD-----------C00 R XX XX  ----------4FB9 NOTE 3 & 4FCB NOTE 4
STATUS WORD--------S00 0 00 00  --------------------------4FBD NOTE 5
ERROR:  ---------------------------------------------------4FC1 NOTE 6
FUNCTION:  ------------------------------------------------4FC4 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------4FBF NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
4FB9H=MSB
4FBAH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4FBBH=MSB
4FBCH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4FBDH=MSB
4FBEH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4FC4H=2ND MSB 4FC6H=LSB
4FC5H=MSB 4FC7H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4FBFH=LSB
4FC0H=MSB

RT 19

```
                                                  00:13:29
                                                  HEX 1553
   DISPLAY RT 19                                     RT 19

ENAB-------------------------------------NO  ----------------4FC8 NOTE 1
RESP BUS---------------------------------NOR ----------------4FD1 NOTE 2
RESP CMD-----------C00 R XX XX  -------------4FC9 NOTE 3 & 4FCB NOTE 4
STATUS WORD--------S00 0 00 00  ---------------------------4FCD NOTE 5
ERROR: -----------------------------------------------------4FD1 NOTE 6
FUNCTION: --------------------------------------------------4FD4 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA: ------------------------------------------------------4FCF NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
4FC9H=MSB
4FCAH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4FCBH=MSB
4FCCH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4FCDH=MSB
4FCEH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4FD4H=2ND MSB 4FD6H=LSB
4FD5H=MSB 4FD7H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4FCFH=LSB
4FD0H=MSB

RT 20

```
                                          00:13:29
                                          HEX 1553
   DISPLAY RT 20                             RT 20

ENAB-------------------------------------NO  ----------------4FD8 NOTE 1
RESP BUS---------------------------------NOR ----------------4FE1 NOTE 2
RESP CMD----------C00 R XX XX  ----------4FD9 NOTE 3 & 4FDB NOTE 4
STATUS WORD-------S00 0 00 00  --------------------------4FDD NOTE 5
ERROR:  ---------------------------------------------------4FE1 NOTE 6
FUNCTION:  ------------------------------------------------4FE4 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------4FDF NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO, 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY
4FD9H=MSB
4FDAH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4FDBH=MSB
4FDCH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4FDDH=MSB
4FDEH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4FE4H=2ND MSB 4FE6H=LSB
4FE5H=MSB 4FE7H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4FDFH=LSB
4FE0H=MSB

RT 21

```
                                          00:13:29
                                          HEX 1553
   DISPLAY RT 21                             RT 21

ENAB-------------------------------------NO  ----------------4FE8 NOTE 1
RESP BUS---------------------------------NOR ----------------4FF1 NOTE 2
RESP CMD-----------C00 R XX XX  ----------4FE9 NOTE 3 & 4FFB NOTE 4
STATUS WORD--------S00 0 00 00  --------------------------4FED NOTE 5
ERROR:  ---------------------------------------------------4FF1 NOTE 6
FUNCTION:  ------------------------------------------------4FF4 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------4FEF NOTE 8
0000 --------0000    0001    0002    0003
0004 --------0004    0005    0006    0007
0008 --------0008    0009    000A    000B
0012 --------000C    000D    000E    000F
0016 --------0010    0011    0012    0013
0020 --------0014    0015    0016    0017
0024 --------0018    0019    001A    001B
0028 --------001C    001D    001E    001F
```

NOTE 1: 00=NO, 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY
4FE9H=MSB
4FEAH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4FEBH=MSB
4FECH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4FEDH=MSB
4FEEH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
4FF4H=2ND MSB 4FF6H=LSB
4FF5H=MSB 4FF7H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4FEFH=LSB
4FF0H=MSB

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 22                                   RT 22

ENAB-------------------------------------NO  ----------------4FF8 NOTE 1
RESP BUS---------------------------------NOR  ---------------5001 NOTE 2
RESP CMD-----------C00 R XX XX  ------------4FF9 NOTE 3 & 4FFB NOTE 4
STATUS WORD--------S00 0 00 00  ----------------------------4FFD NOTE 5
ERROR:  ----------------------------------------------------5001 NOTE 6
FUNCTION:  -------------------------------------------------5004 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------------------------4FEF NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO, 80H=YES

NOTE 2: FORMAT IS BBBEEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY
4FF9H=MSB
4FFAH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
4FFBH=MSB
4FFCH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
4FFDH=MSB
4FFEH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
5004H=2ND MSB 5006H=LSB
5005H=MSB 5Q07H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
4FFFH=LSB 5000H=MSB

RT 23

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 23                                   RT 23

ENAB-------------------------------------NO  ----------------5008 NOTE 1
RESP BUS---------------------------------NOR ----------------5011 NOTE 2
RESP CMD----------C00 R XX XX  ----------5009 NOTE 3 & 500B NOTE 4
STATUS WORD-------S00 0 00 00  ----------------------------500D NOTE 5
ERROR: -----------------------------------------------------5011 NOTE 6
FUNCTION: --------------------------------------------------5014 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA: ------------------------------------------------------500F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO, 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY
5009H=MSB
500AH=LSB

NOTE 4. SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'
500BH=MSB
500CH=LSB

NOTE 5. SIXTEEN BIT BINARY FIELD.
500DH=MSB
500EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
5014H=2ND MSB 5016H=LSB
5015H=MSB 5017H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
500FH=LSB
5010H=MSB

RT 24

```
                                        00:13:29
                                        HEX 1553
   DISPLAY RT 24                           RT 24

ENAB-------------------------------------NO  ---------------  ---5018 NOTE 1
RESP BUS---------------------------------NOR  ---------------  5021 NOTE 2
RESP CMD-------------C00 R XX XX  ------------5019 NOTE 3 & 501B NOTE 4
STATUS WORD----------S00 0 00 00  ---------------------------  501D NOTE 5
ERROR:  ----------------------------------------------------  5021 NOTE 6
FUNCTION:  -------------------------------------------------  5024 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------------------------  501F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
5019H=MSB
501AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
501BH=MSB
501CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
501DH=MSB
501EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
5024H=2ND MSB 5026H=LSB
5025H=MSB 5027H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
501FH=LSB
5020H=MSB

```
                                          00:13:29
                                          HEX 1553
   DISPLAY RT 25                             RT 25

ENAB-------------------------------------NO  ----------------5028 NOTE 1
RESP BUS---------------------------------NOR ----------------5031 NOTE 2
RESP CMD----------C00 R XX XX  ----------5029 NOTE 3 & 502B NOTE 4
STATUS WORD-------S00 0 00 00  ----------------------------502D NOTE 5
ERROR:  ----------------------------------------------------5031 NOTE 6
FUNCTION:  -------------------------------------------------5034 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------------------------502F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD BINARY.
5029H=MSB
502AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
502BH=MSB
502CH=LSB

NOTE 5. SIXTEEN BIT BINARY FIELD.
502DH=MSB
502EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
5034H=2ND MSB 5036H=LSB
5035H=MSB 5037H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
502FH=LSB
5030H=MSB

RT 26

```
                                        00:13:29
                                        HEX 1553
   DISPLAY RT 26                           RT 26

ENAB-------------------------------------NO  ----------------5048 NOTE 1
RESP BUS---------------------------------NOR ----------------5041 NOTE 2
RESP CMD-----------C00 R XX XX  ----------5039 NOTE 3 & 503B NOTE 4
STATUS WORD--------S00 0 00 00  ----------503D NOTE 5
ERROR:  ----------------------------------5041 NOTE 6
FUNCTION:  -------------------------------5044 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------503F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO, 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
5039H=MSB
503AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
503BH=MSB
503CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
502DH=MSB
502EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: TWENTY-FOUR BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
5044H=2ND MSB 5046H=LSB
5045H=MSB 5047H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
503FH=LSB
5040H=MSB

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 27                                   RT 27

ENAB-------------------------------------NO  ----------------5048 NOTE 1
RESP BUS---------------------------------NOR ----------------5051 NOTE 2
RESP CMD----------C00 R XX XX  ----------5049 NOTE 3 & 504B NOTE 4
STATUS WORD-------S00 0 00 00  ----------------------------504D NOTE 5
ERROR:  ----------------------------------------------------5051 NOTE 6
FUNCTION:  -------------------------------------------------5054 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------------------------504F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
5049H=MSB
504AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'
504BH=MSB
504CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
504DH=MSB
504EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
5054H=2ND MSB 5056H=LSB
5055H=MSB 5057H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
504FH=LSB
5050H=MSB

RT 28

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 28                                   RT 28

ENAB-------------------------------------NO  ----------------5058 NOTE 1
RESP BUS---------------------------------NOR  ---------------5061 NOTE 2
RESP CMD-----------C00 R XX XX  ------------5059 NOTE 3 & 505B NOTE 4
STATUS WORD--------S00 0 00 00  ----------------------------505D NOTE 5
ERROR:  ----------------------------------------------------5061 NOTE 6
FUNCTION:  -------------------------------------------------5064 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  -----------------------------------------------------505F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO, 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
5059H=MSB
505AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
505BH=MSB
505CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
505DH=MSB
505EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
5064H=2ND MSB 5066H=LSB
5065H=MSB 5067H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
505FH=LSB
5060H=MSB

```
                                          00:13:29
                                          HEX 1553
   DISPLAY RT 29                             RT 29

ENAB-------------------------------------NO  ----------------5068 NOTE 1
RESP BUS---------------------------------NOR ----------------5071 NOTE 2
RESP CMD----------C00 R XX XX  ------------5069 NOTE 3 & 506B NOTE 4
STATUS WORD-------500 0 00 00  ----------------------------506D NOTE 5
ERROR:  ---------------------------------------------------5071 NOTE 6
FUNCTION:  ------------------------------------------------5074 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------506F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO, 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY
5069H=MSB
506AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
506BH=MSB
506CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
506DH=MSB
506EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
5074H=2ND MSB 5076H=LSB
5075H=MSB 5077H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
506FH=LSB
5070H=MSB

RT 30

```
                                        00:13:29
                                        HEX 1553
   DISPLAY RT 30                           RT 30

ENAB-------------------------------------NO  ----------------5078 NOTE 1
RESP BUS---------------------------------NOR ----------------5081 NOTE 2
RESP CMD----------C00 R XX XX  ----------5079 NOTE 3 & 507B NOTE 4
STATUS WORD-------S00 0 00 00  ----------------------------507D NOTE 5
ERROR:  ---------------------------------------------------5081 NOTE 6
FUNCTION:  ------------------------------------------------5084 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------507F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY
5079H=MSB
507AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
507BH=MSB
507CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
507DH=MSB
507EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3)..

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
5084H=2ND MSB 5086H=LSB
5085H=MSB 5087H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
507FH=LSB
5080H=MSB

RT 31

```
                                                00:13:29
                                                HEX 1553
   DISPLAY RT 31                                   RT 31

ENAB-------------------------------------NO  ----------------5088 NOTE 1
RESP BUS---------------------------------NOR ----------------5091 NOTE 2
RESP CMD----------C00 R XX XX  ----------5089 NOTE 3 & 508B NOTE 4
STATUS WORD-------500 0 00 00  --------------------------508D NOTE 5
ERROR:  ---------------------------------------------------5091 NOTE 6
FUNCTION:  ------------------------------------------------5094 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:  ----------------------------------------------------508F NOTE 8
0000 --------0000     0001     0002     0003
0004 --------0004     0005     0006     0007
0008 --------0008     0009     000A     000B
0012 --------000C     000D     000E     000F
0016 --------0010     0011     0012     0013
0020 --------0014     0015     0016     0017
0024 --------0018     0019     001A     001B
0028 --------001C     001D     001E     001F
```

NOTE 1: 00=NO. 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
5089H=MSB
508AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'
508BH=MSB
508CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
508DH=MSB
508EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
5094H=2ND MSB 5096H=LSB
5095H=MSB 5097H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX.
508FH=LSB
5090H=MSB

```
                                        00:13:29
                                        HEX 1553
                                           RT 32
   DISPLAY RT 32

ENAB.....NO.....5098 NOTE 1
RESP BUS.....NOR.....50A1 NOTE 2
RESP CMD.....C00 R XX XX.....5099 NOTE 3 & 509B NOTE 4
STATUS WORD.....500 0 00 00.....509D NOTE 5
ERROR:.....50A1 NOTE 6
FUNCTION:.....50A4 NOTE 7
 0000 0000 0000 0000 0000 0000 0000 0000
DATA:.....509F NOTE 8
0000 .....0000     0001     0002     0003
0004 .....0004     0005     0006     0007
0008 .....0008     0009     000A     000B
0012 .....000C     000D     000E     000F
0016 .....0010     0011     0012     0013
0020 .....0014     0015     0016     0017
0024 .....0018     0019     001A     001B
0028 .....001C     001D     001E     001F
```

NOTE 1: 00=NO, 80H=YES

NOTE 2: FORMAT IS BBBEEEEE. BBB IS BUS SELECT. SEE NOTE 6 FOR EEEEE. BBB DESCRTIPTION: 000=NOR 010=INV 100=PRI 110=SEC

NOTE 3: SIXTEEN BIT RESPONSE COMMAND FIELD. BINARY.
5099H=MSB
509AH=LSB

NOTE 4: SIXTEEN BIT MASK FIELD. BINARY. EACH BIT SET TO A ZERO WILL CAUSE THE CORRESPONDING BIT IN THE RESPONSE COMMAND FIELD TO BE A 'DONT CARE'.
509BH=MSB
509CH=LSB

NOTE 5: SIXTEEN BIT BINARY FIELD.
509DH=MSB
509EH=LSB

NOTE 6: REFER TO THE BC/RT ERROR TABLE (Table 4-3).

NOTE 7: THIRTY-TWO BIT BINARY FIELD. EACH BIT IS REPRESENTED BY A ZERO OR ONE ON THE FUNCTION LINE.
50A4H=2ND MSB 50A6H=LSB
50A5H=MSB 50A7H=2ND LSB

NOTE 8: SIXTEEN BIT BINARY FIELD. 667 MAX
509FH=LSB
50A0H=MSB

TABLE 4-3. BC/RT ERRORS (Cont'd)

ZERO XING FIXED WORD XX XING YY	4CB1H=BBB10001 4CB2H=WORD # (20H MAX) 4CB3H=XING # (22H MAX)
ZERO XING RIPPLE THRU WORD XX	4CB1H=BBB10010 4CB2H=WORD # (20H MAX) 4CB3H=0
RESPONSE TIME + XX	4CB1H=BBB10011 4CB2H=0 4CB3H=TIME. MUST BE FROM 4-22H IN EVEN STEPS. NOTE: USED BY RT'S ONLY.
XMIT ON BOTH BUSSES	4CB1H=BBB10100 4CB2H=0 4CB3H=0
NOISE GATE XX	4CB1H=BBB10101 4CB2H=0 4CB3H=TIME. (0AH MAX)

4.5.2. Reading the BAP Statistics Table

The Background Activity Processor (BAP) in the SBA 100F runs continuously to monitor the performance of the bus under test. When control is assigned to the RS-232 or GPIB ports, the update of the Terminal page stops, making it difficult to determine the status of the bus from a host computer

The following explanation describes a method of reading the BAP Statistic Table and transferring this information to a host computer The host may then continuously monitor activity, with no interruption.

Selecting the BAP RAM is done by the following:

The BAP RAM is mapped into the SBA's address space at A00H to BFFFH. This area of RAM is shared by a Selective Activity Monitor, so a special selection technique must be used to ensure communication with the proper device. To access the BAP RAM a non-zero value must be written to I/O port 80H prior to the first RAM access. This must be done using the SBA's GOSUB XXXX command. The following routine will select the BAP RAM when executed using the GOSUB XXXX command.

Once you have selected the BAP RAM, you can read the memory and determine the current total error count and the specific types of errors. An alternative method for achieving the same end, without handling the shared BAP/SAM RAM, is to display the Terminal or Stack page and read the data from the CRT display memory region, located from 8000H to 9400H. Both of these methods are described in the following sections. The first method for reading memory pages to record activity monitoring involves reading the BAP RAM. The second method presented is reading the CRT memory.

4.5.2.1. Reading The BAP Memory

The first step is to access the BAP memory page. This is done by saving a non-zero number to I/O port 80H. prior to the RAM access This is done as follows:

HEX	MNEMONIC	COMMENTS
3EFF	MVI A.255	SELECT BAP RAM
D380	OUT 80H	
C9	RET	BACK TO CALLER

Reading the total error count from the BAP is accomplished by accessing the memory beginning at BFE8H to BFEB (LSB-> MSB). The BAP maintains a 32-bit binary count of all errors occurring on the bus. This count is available to the system beginning at address BFE8H. The format of this count is as follows:

BFE8H	LEAST SIGNIFICANT BYTE
BFE9H	2ND LEAST SIGNIFICANT BYTE
BFEAH	3RD LEAST SIGNIFICANT BYTE
BFEBH	MOST SIGNIFICANT BYTE

This field may be cleared by executing the SBA's display terminal clear command.

Determining the specific errors encountered requires the decoding of the table of statistics maintained for the first 128 unique commands it sees on the bus. Each command appearing on the bus will have a 12 byte statistics block assigned to it. This statistics table begins at the address 0A0CH and ends at 0A6CH. The format for the first entry is described below. These tables are arranged in the order of first seen first entered and not in the ascending order shown on the terminal page. These tables may be transferred to the host using the Save Memory to GPIB/RS-232 command

ADDRESS	FUNCTION	FORMAT
A00CH	COMMAND WORD	LSB
A00DH		MSB
A00EH	32 BIT BINARY	LSB
A00FH	COMMAND OCCURENCE	2ND LSB
A010H	COUNT	3RD LSB
A011H		MSB
A012H	FLAG WORD #1	LSB - SEE BELOW
A013H		MSB
A014H	FLAG WORD #2	LSB - SEE BELOW
A015H		MSB
0A016H	16 BIT BINARY	LSB
0A017H	ERROR COUNT	MSB

FLAG WORD #1	FLAG WORD #2
D15= 1=ERROR, 0=NO ERROR	D15=PRIMARY BUS
D14= BITERR	D14=SECONDARY BUS
D13= LOBIT	D13=RT TO RT XFER
D12= HIBIT	D12 THRU D0 UNUSED
D11= PARITY	
D10= TWOBUS	
D09= SYNC	
D08= GAP	
D07= HIWORD	
D06= LOWORD	
D05= NORESP	
D04= RMODE	
D03= DELAY	
D02= EARLY	
D01= MEBIT	
D00= STABIT	

NOTE 1

Invalid blocks are indicated by D15 and D14 in Flag Word #2 being clear.

Note 2:

No error conditions are indicated by D15 in Flag Word #1 being clear or an error count = 0

Note 3:

The active state of all bits is 1 unless stated otherwise.

4.5.2.2. Reading The CRT Memory

The second method for reading background activity in memory involves looking at the actual CRT memory, rather than visually determining its correctness. This method is an aid for automated test situations. The first step is to display the page in question. For example, to verify an entry on the Stack page, display the Stack page and check the appropriate memory location to verify the entry. The Stack page and the Terminal Activity page are used for examples of memory reading of the CRT on the pages which follow.

Example:

```
DISPLAY TERMINAL 07
TEST MEMORY 8528 = "NORESP"
EXECUTE JUMP TO KEY 010 IF TRUE
```

Note that when reading the CRT memory, the MSB is high (begins with a one). Before you read the ASCII table in Appendix E, you must first zero out the first character.

4.5.2.3. Terminal Page CRT Memory Map

The example given here is for line 1 of the Terminal page, which begins at 8500H. Each line begins on a multiple of 100, meaning that each subsequent line begins at the previous line + 100. The Terminal page begins at 8500H and ends at 8F00. The total error count is found beginning at location 9213H. Since the Terminal page display may take up more than one screen display, if you want to see the CRT memory of a specific terminal, enter the command DISPLAY TERMINAL XX, where XX = the terminal number you want. The first line will contain the information of the designated terminal.

To read line 1 of the Terminal page in CRT memory, see below:

TERMINAL PAGE FORMAT

```
-------------------------------------------------------------------------
RT  TERM  SUB  WC  BUS  CMDS  ERRS  ERRORS
-------------------------------------------------------------------------
```

The CRT memory looks like this:

RT	: 8502 = R, if RT to RT transfer
TERM	: 8503, 8504
T or R	: 8509
SUB	: 850B, 850C
WC	: 8511, 8512
BUS	: 8517, 8518, 8519
CMDS	: 851B, 851C, 851D, 851E, 851F
ERRS	: 8521, 8522, 852A, 852B, 852C, 852D

Line 2 begins at 8700. Location 8600 will contain entries in the ERRORS field only. If there are less than two errors, this line will contain no useful information.

To read the errors detected (ERRS column) for terminal 14 (as an example), perform the following:

1. Press the D, T, 1, and 4 keys. The data for terminal 14 will display on the CRT starting at line one.
2. Read the memory (via a host with a SAVE MEMORY) and translate as follows:

ERRORS MEMORY:	8528H	8529	852A	852B	852C	852D
HEX VALUE:	CE	CF	D2	C5	D3	D0
BINARY VALUE:	11001110	11001111	11010010	11000101	11010011	11010000
ZERO OUT MSB:	01001110	01001111	01010010	01000101	01010011	01010000
RESULT DEC:	78	79	82	69	83	80
ASCII CHAR:	N	O	R	E	S	P

The error detected was a NORESP.

Lines 3 - 12 begin at locations 8800 - 9100, beginning at intervals of 100.

4.5.2.4. Stack Page CRT Memory Map

The example which follows is for line 1 of the Stack page. which begins at 8600H. Each line begins on a multiple of 100 Line 1 of the Stack page begins at 8600. line 2 at 8700. up to 9100 for the first screen display of stack data. You can increment the Stack page display and re-read the CRT memory. but the CRT memory contains a maximum of 12 lines of stack data.

To read line 1 of the Stack page. see below:

STACK PAGE FORMAT.

```
-------------------------------------------------------------------
ADDR  BUS  ERRORS  WN  STACK DATA
-------------------------------------------------------------------
```

The CRT memory looks like this:

ADDR	: 8603. 8604. 8605. 8606
BUS	: 8608. 8609. 860A. 860B
ERRORS	: 860D. 860E. 860A. 860B
WN	: 8614. 8615
COMMAND	: 8618 = N. if new command R. if RT-RT transfer
STACK DATA	: 8619. 862E

Line 2 begins at 8700. lines 3 - 12 begin at intervals of 100.

For example. to read the stack data for lines 38. 39. and 40. use the command DISPLAY STACK 38 to display line 38. Then read memory locations 8600. 8700. and 8800.

4.5.2.4. Stack Page CRT Mo

The examp... follows is ...
8600H Each line begins on a mul...
begins at 8600 ...ine 2 at 870...
data You can increment the S...
memory but the CRT memory co...

To read line 1 of the Stack page ...

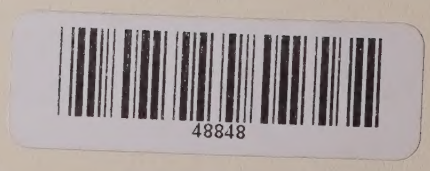
STACK PAGE "OMMA"

ADDR BUS ERRORS WN STAC...

The CRT memory looks li...
ADDR
BUS
ERRORS
...N
COMMAND
STACK DATA

Line 2 begins at 8700 lines 3 ...

For example, to re... the Stack Da...
mand DISPLAY STACK ...
8600 8700 and 8800


48848